SLED 10 » GNOME 2.16 » Ulteo » Compiz

LXF DVD внутри!

LINUX FORMAT

Главное в мире Linux

Не пропустите!

SUSE LINUX Enterprise Desktop 10

полная версия! с. 136 »

ноябрь 2006 № 11 (85)

Сокрушительный стек

Novell представляет новую концепцию Linux для бизнеса – готовы ли мы оказаться в ее руках?

Get Things Done!

- » Защити свой сервер с. 82
- » Формы и отчеты в OOo с. 74
- » Программируем роботов с. 86
- » Изучаем Inkscape 0.44 с. 78
- » Уникальный рабочий стол с. 90
- » Верстаем в Scribus с. 60
- » Нелинейный видеомонтаж в Cinelerra с. 54

Замечательный дистрибутив
Гаэль Дюваль поделился с нами планами по захвату мира

Java без слез
Пишите код быстрее и лучше – мы сравнили самые популярные IDE для Java!

4 607130 820452

« По-моему, бесполезно пытаться превратить Linux в настольную ОС »

Кристиан ван дер Влиет с. 36



К Вашим услугам...

Наблюдая за развитием проекта One Laptop Per Child, мы поинтересовались у Команды LXF: «А что еще, по-вашему, должно стоить $100»?



Пол Хадсон
Linux Format. Уж тогда нам точно повысят зарплату.



Грэм Моррисон
Сто фунтов стерлингов. К счастью, производители пока следуют этой традиции.



Майк Сондерс
Устройства Судного Дня. Чтобы заглянуть в Argos – и получить комплект для уничтожения мира с бесплатной доставкой.



Эфрейн Эрнандес-Мендоса
Я бы отдал столько за итальянскую стрижку. Если добавят Ferrari.



Ребекка Смолли
Кросс-атлантический и кросс-тихоокеанский перелет на одномачтовой лодке. Дешевые авиабилеты должны умереть!



Эндрью Грегори
Ядерное оружие. По боеголовке на церковь/мечеть/синагогу – и вот он, долгожданный мир во всем мире.



Джонатан Остин
Ноутбуки от OLPC. Я слышал, они подорожали до $140.



Нейл Ботвик
Все! Подумайте, сколько времени вы тратите, выбирая место, где можно купить что-нибудь по самой выгодной цене.



Д-р Крис Браун
Пожизненная подписка на Linux Format, я думаю.



Дэвид Картрайт
Пять пятифунтовых купюр. Таким образом, я получу в два раза больше американского пива за те же деньги.



Ричард Коббет
Ну... $200. Конечно, начнется кошмарная гиперинфляция, но пока суд да дело – повеселимся.



Энди Хадсон
Доступ во все закоулки Googleplex. Хочу увидеть все своими глазами.



Обнаружено новое устройство

Мы в редакции Linux Format частенько получаем письма от людей, которые просят помочь им настроить работу какого-либо устройства в Linux (да, видеокарты ATI по-прежнему лидируют). Возникает резонный вопрос: почему, даже несмотря на то, что число поддерживаемых Linux устройств растет и растет, таких просьб не становится меньше?

Ответ прост – время идет, но процедура установки остается прежней: зайдите туда-то, скачайте то-то, установите abc и def. Речь не о том, что это бывает сложно или неудобно – но как человек, впервые установивший Linux (или того страшнее – получивший его вместе с новым компьютером), догадается, что он должен сходить на такой-то сайт или подключить сторонний репозиторий? Естественно, помыкавшись 5-10-15-20 минут, бедный пользователь со словами: «Ну его, этот Linux» поставит привычный Windows XP, который не стесняется показать окошко «Обнаружено новое устройство», даже если ничего путного с этим устройством поделать не может. Почему бы производителям дистрибутивов не сделать еще один шаг навстречу новичкам? Не хотите включать закрытые драйвера ATI – не надо, но, определив видеоадаптер как Radeon 9600, не поленитесь показать окно с сообщением: «Для поддержки вашей видеокарты в Linux нужно загрузить драйвер. Обратитесь по адресу: ...». Можно пойти еще дальше и предоставлять сводку вида: «Вот это поддерживается из коробки, для этого – скачайте дополнительный пакет из репозитория R, это – поддерживается сообществом, а это – просто не работает». У производителей накоплен значительный опыт настройки различных устройств и они обычно находятся в курсе всех дел – так почему бы не поделиться этой информацией с пользователем? LXF

Валентин Синицын » Главный редактор info@linuxformat.ru

Миссия журнала



- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru

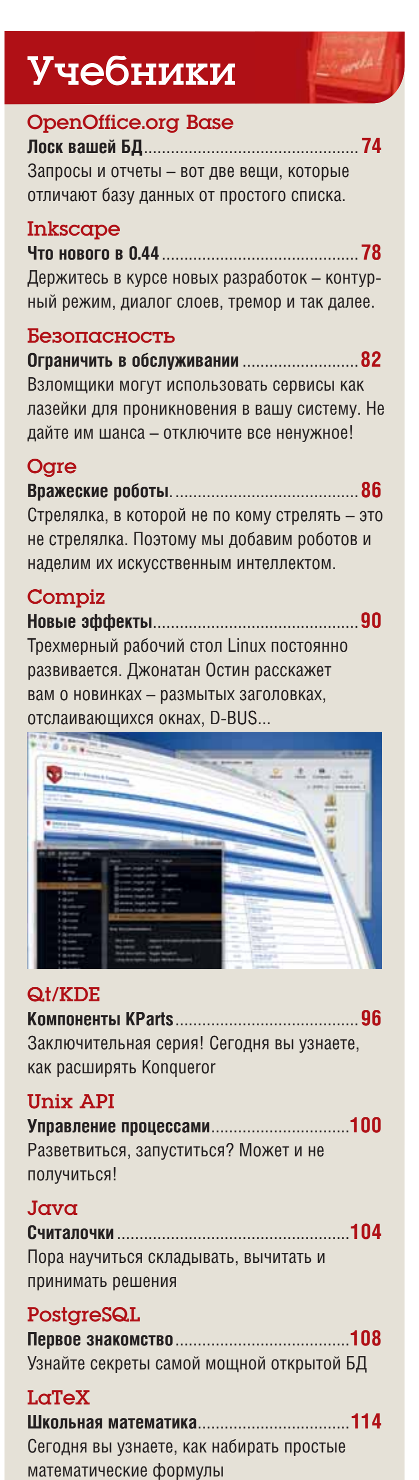
› **Адрес редакции:** Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54.
› **Телефон редакции:** (812) 717-00-37. Дополнительная информация на стр.142



Содержание

Весь номер – прямо как на ладони: приятного чтения!

Учебники





LXF DVD85

SLED 10
Корпоративный рабочий стол от Novell – лучшее от SUSE плюс промышленный лоск и стабильность. Идеальный дистрибутив для домашних и офисных рабочих станций.

ReactOS 0.3.0
Вам Linux – друг, но Windows дороже? Попробуйте открытый клон Microsoft Windows, который становится все популярнее с каждым днем.

Glade 3.0
Создавать интерфейс приложений Gnome и GTK стало проще с выходом новой версии Glade. Теперь с контекстной помощью и панелями инструментов!

Wormux
Черви вернулись! Хватайте базуку – и наслаждайтесь классической многопользовательской игрой.



› SLED 10 – идеален для дома и офиса

Что за штука... CSS 3?

Откройте для себя мощь семантического web!
с. 40

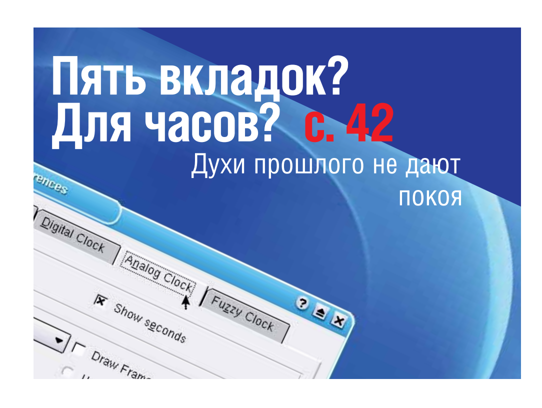




LXFHotPicks



› Может ли заставка учить японскому?







Подпишись на Linux Format и сэкономь!



LXF DVD внутри!

См. страницу 136



Спецрепортаж

Сокрушительный стек

Что делает Novell, чтобы потеснить Red Hat на предприятиях с. 28

А также...



Постоянные рубрики



› Ладислав рассуждает о реванше Fedora.

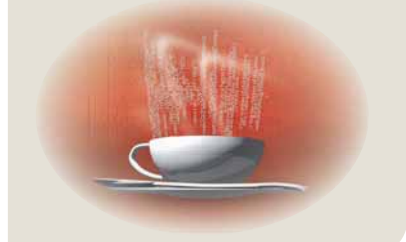
Обзоры





› Это не Т-1000 в отпуске – это Blender 2.42 за работой.

Сравнение: Java IDE



Новости



ГЛАВНЫЕ НОВОСТИ: Корпорация Mozilla спорит с Debian и выпускает Firefox 2.0 » Linux становится системой реального времени » Portland 1.0 » Parallels Workstation 2.2 » YDL для Sony PS3 » Oracle Unbreakable Linux

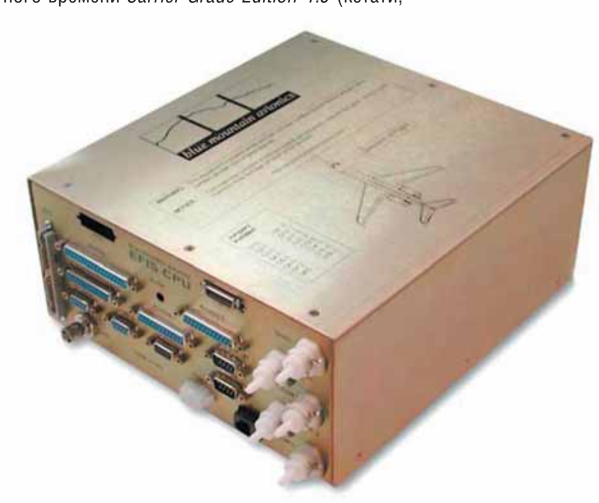
Новости подготовил Илья Шпаньков

В погоне за временем

По сообщению компании TimeSys, являющейся одним из ведущих разработчиков ПО для встроенных систем на основе Linux, начиная с версии 2.6.18 ядро Linux включает в себя поддержку технологии реального времени. Ранее аналогичные средства добавлялись в ядро в виде патчей. Включение подобной технологии в основную ветку ядра в наибольшей степени стало возможным благодаря одному из ведущих программистов компании – Томасу Глейкснеру [Thomas Gleixner], который является основным автором подсистемы высокоточного таймера (*hrtimer – «high-resolution timer»*), входящего в ядро, а также внес большой вклад в отладку системы приоритета прерываний, разрабатываемой Инго Мольнаром [Ingo Molnar] из Red Hat. В общей сложности список изменений новой версии ядра содержит 136 добавлений авторства Глейкснера и 143 – Мольнара.

Надо сказать, что подобная функциональность уже давно была востребована разработчиками встроенных систем, требующих корректной обработки ядром Linux процессов, работающих в режиме реального времени, с точностью до миллисекунды. До настоящего момента программистам приходилось самостоятельно «подгонять» ядро для работы в устройствах, предназначенных для выполнения критичных по времени задач, но начиная с Linux 2.6.18, они получили практически готовый к использованию программный продукт. Особой популярностью подобные системы пользуются у разработчиков авиационного оборудования. В качестве примера можно вспомнить компанию FSMLabs, которая использовала Linux реального времени при создании электронной полетной информационной системы (electronic flight information system – EFIS) для экспериментальных моделей самолетов. Эта система позволяет пилотам визуально ориентироваться по трехмерному изображению поверхности Земли, расположенному на экране монитора. На применение в авиационной технике рассчитана и операционная система реального времени Sand Linux, разрабатываемая российской компанией Cyber System Development (http://www.cybsysdev.com/).

Между тем, высокоточное ПО нужно не только авиаторам. Разрабатываемая компанией MontaVista операционная система реального времени *Carrier Grade Edition 4.0* (кстати, уже построенная на ядре версии 2.6.18) принимается «на вооружение» в Дании для обеспечения коммуникационных потребностей военных, пожарных, медиков и полицейских. В то же самое время компания Novell объявляет о выпуске SUSE Linux Enterprise Real Time (SLERT), предназначенной для любых потребителей, нуждающихся в системах реального времени. Подобная активность объясняется все увеличивающимся спросом на подобное ПО, и новое ядро с уникальными возможностями позволит открытому программному обеспечению еще больше укрепиться на данном рынке.





Новости короткой строкой

» Adobe Labs анонсировала выход бета-версии *Flash Player 9* для Linux. Финальная версия ожидается в начале 2007 года.

» Вышла очередная стабильная версия популярной графической среды KDE 3.5.5.

» Пакет офисных приложений *OpenOffice.org* обновился до версии 2.0.4, одновременно вниманию пользователей представлена одноименная сборка от компании «Инфра-ресурс», оптимизированная для русскоговорящих пользователей.

» Компания Panda выпустила бета-версию *Panda DesktopSecure* для Linux.

» Администрация Мюнхена планирует перевести к 2008 году 80% своих компьютеров на свободное ПО, затратив на эти цели около 44,5 млн. долларов США.

Portland: Объять необъятное

Совместный проект Open Source Development Labs (OSDL) и *Freedesktop.org*, призванный облегчить разработку приложений, способных одинаково хорошо функционировать в средах KDE и GNOME, наконец получил статус стабильной версии под номером 1.0. Название «Portland» (напоминающее о городе, в котором в конце 2005 года и родилась идея проекта) «скрывает» в себе два основных компонента: комплект утилит *xdg-utils*, работающих из командной строки, и набор интерфейсов DAPI, используемый приложениями при функционировании в «неродной» графической среде. Несмотря на стабильный номер версии *Portland*, DAPI пока по-прежнему остается на стадии Technology Preview, а все «бремя славы» принимает на себя набор утилит. Основная задача этих небольших программ – унифицирование для различных графических сред операций по установке и удалению пунктов меню и иконок на рабочем столе, обработке неизвестных типов файлов, запуску предпочтительных приложений, выполнению программ с правами администратора, управлению работой хранителя экрана. Таким образом, включение поддержки *xdg-utils* и DAPI в приложения, разрабатываемые для одной из двух популярных оконных сред, позволяет избавиться от массы проблем (и установки дополнительных библиотек) при работе в любой из них.

Проект оказался востребованным и сегодня о его перспективности заявляют такие компании, как *Trolltech*, включившие поддержку *Portland* в новую версию библиотеки *Qt 4.2*, а разработчики дистрибутивов Red Flag и *Xandros* планируют ввести *Portland* в очередные стабильные релизы своих программных продуктов. Стоит отметить также, что многие популярные дистрибутивы уже не только содержат, но и активно применяют некоторые компоненты из состава универсальных утилит. Подобная популярность молодого проекта позволила аналитикам пересмотреть свои прогнозы на будущее всех GNU/Linux систем в целом и предположить, что унифицирование пользовательского интерфейса в различных графических средах будет способствовать росту числа поклонников Linux и в 2008 году общий денежный оборот на рынке свободных настольных операционных систем составит не менее 10 миллиардов долларов.

http://portland.freedesktop.org/wiki/

Свобода обретает границы?



Пока идут жаркие споры вокруг третьей редакции общественной лицензии GPL, в свободном сообществе родился новый конфликт юридического характера, связанный с браузером *Firefox*. Фонд Mozilla зарегистрировал названия своих программных продуктов и соответствующие им иконки как товарные знаки, одновременно запретив их раздельное использование. Также появились изменения и в системе взаимоотношений фонда с разработчиками свободно распространяемых дистрибутивов GNU/Linux: теперь все вносимые, например, в *Firefox* изменения (производимые практически каждым дистрибьютором) должны проходить тестирование программистами Фонда – только после этого авторы дистрибутивов получают право на использование названия и иконки популярного браузера. По словам представителей Фонда, данный шаг обусловлен заботой о конечном пользователе, который таким образом получит гарантии качества и заявленной функциональности *Firefox*.

Между тем, разработчики Debian усмотрели в подобных действиях противоречие рекомендациям Debian Free Software Guidelines (DFSG), используемым для определения пригодности программного обеспечения к включению в состав дистрибутива. В Сети разгорелись жаркие споры по данному вопросу, но до сих пор какого-либо четкого решения нет ни у одной из сторон. Мнения высказываются самые различные: кто-то советует проекту Debian смириться и выполнить предписания Фонда, а кто-то призывает перенести браузер в категорию несвободного ПО (non-free) или вообще отказаться от его использования. Многие также предлагают помещать *Firefox* (код которого по-прежнему остается свободным) в дистрибутив под другим именем – Iceweasel («Горностай»), тем самым избежав нарушения предписаний Фонда Mozilla, но подобное решение значительно усложняет сопровождение Debian. Каким будет финальное решение проблемы – покажет время.

gnuzilla.gnu.org

Parallels Workstation 2.2 для Windows и Linux

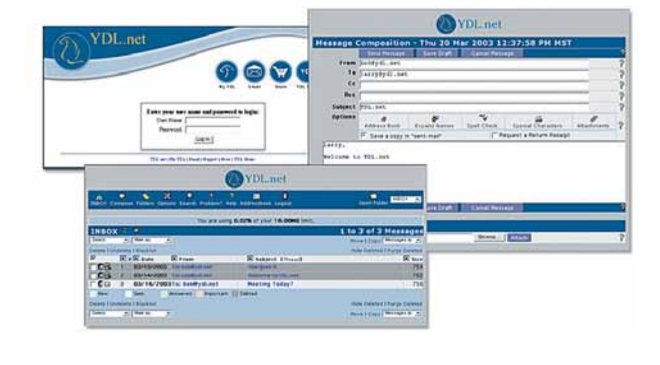


Компания Parallels объявила о готовности новой версии виртуальной машины Parallels Workstation. Подобно *VMWare*, данное ПО предназначено для запуска нескольких операционных систем на одном компьютере. С помощью *Workstation* можно «поселить по соседству» такие системы, как Windows, Linux, *Solaris*, *FreeBSD*, *NetBSD*, *OpenBSD*, *OS/2*, eComStation или DOS. Правда, запустить саму виртуальную машину удастся лишь на компьютерах под управлением Windows 2000, XP или 2003, Linux или Mac OS X (в последнем случае потребуется отдельный продукт). Очередная версия *Workstation* получила такие новшества, как возможность запуска всех тестовых Beta и RC-версий Windows Vista, улучшенная поддержка Intel Virtualization Technology и AMD Secure Virtual Machine Technology (SVM), более стабильную работу с USB-интерфейсом, полную поддержку Unicode. Также значительные улучшения коснулись работы с сетью (в т.ч. и беспроводной), записи и воспроизведения звуковых файлов. Отдельно можно отметить появление утилиты *Shared Folders*, позволяющей сохранять данные в каталогах, доступных из всех установленных с помощью Workstation операционных систем [для гостевых ОС семейства Windows, – прим. ред.]. В отличие от *VMWare Workstation*, стоимость одной лицензии на Parallels Workstation составляет всего $50, правда, бесплатный «Parallels Player» отсутствует.

www.parallels.com



Играть в Linux – это просто



По сообщению компании Terra Soft, во исполнение заключенного с корпорацией Sony Computer Entertainment договора, подготовлена новая версия операционной системы *Yellow Dog Linux 5.0*, предназначенная для установки на игровые приставки *PlayStation 3*. Система поступит в продажу отдельно, а ее установка на приставку будет производиться пользователями самостоятельно. Одним из основных плюсов использования YDL в качестве платформы является тот факт, что после ее установки игровая консоль превращается практически в полноценный компьютер со всеми соответствующими функциональными возможностями, включая работу во всемирной сети и просмотр web-страниц, сохраняя при этом свои развлекательные функции. В состав операционной системы, построенной на базе Fedora Core 5, вошли такие компоненты, как ядро Linux версии 2.6.16, *gcc 4.1.1* и *glibc 2.4*, Cell SDK 1.1, *OpenOffice.org* версии 2.0.2, *FireFox 1.5.0*, *Thunderbird 1.5.0* и Nautilus *2.14*. Также нужно отметить, что при создании данного дистрибутива разработчики тесно сотрудничали с компанией Enlightenment, результатом чего стала новая стильная версия графической оболочки E17.



Oracle Linux – новый поворот

Интерес корпорации *Oracle* к свободным операционным системам и неприкрытое желание использовать их возможности в своей бизнес-практике уже давно обсуждается на страницах различных изданий. Наконец, ситуация прояснилась, и сегодня можно говорить о новом прецеденте в области взаимодействия коммерческих компаний со свободным ПО, и начало ему положили маркетологи *Oracle*. В частности, представители всемирно известной корпорации объявили о начале проекта под названием *Oracle* Unbreakable Linux, в рамках которого пользователям операционных систем, разрабатываемых компанией Red Hat, будет предоставляться полноценная техническая поддержка, но по вдвое меньшей цене, чем у производителя. В результате, акции Red Hat резко упали в цене (почти на 30%). Кроме обеспечения технической поддержки, *Oracle* планирует выпустить собственный клон операционной системы Red Hat Linux, который, по словам инициаторов нового проекта, будет более отлаженным и стабильным, чем оригинал. Подобные планы, а также лучшее по сравнению с Red Hat финансовое положение корпорации, дают аналитикам повод предсказать дальнейшее снижение стоимости акций американской компании и даже предположить возможность поглощения Red Hat рыночным монстром *Oracle*.

Между тем, представители Red Hat не проявляют признаков паники и вполне спокойно относятся к новой инициативе *Oracle*. Например, генеральный директор Red Hat Мэтью Шулик [Matthew Szulik] в своем интервью пояснил, что подобный шаг гиганта софтверной индустрии никак не скажется на планах Red Hat и стоимости услуг компании, и даже внесет дополнительный фактор здоровой конкуренции на рынок свободных операционных систем, и уж тем более не приведет к продаже компании, являющейся одним из лидеров на рынке свободного ПО. Впрочем, спокойствие руководителей Red Hat не мешает многочисленным пользователям свободных систем высказывать свое возмущение инициативами *Oracle*, справедливо усматривая в них признак использования разрабатываемых сообществом свободно распространяемых программных продуктов в корыстных целях.

www.oracle.com/technologies/linux

Встречайте Firefox 2.0

Фонд Mozilla объявил о выходе новой версии браузера *Firefox 2.0*. Значительное изменение номера говорит о том, что разработчики оценивают изменения в коде популярного свободного web-обозревателя как серьезные. И, надо сказать, им есть, что предложить пользователям своего браузера. Особое внимание авторы уделили дальнейшему совершенствованию системы вкладок, используемой для одновременного отображения нескольких страниц. Теперь каждая вкладка обладает индивидуальной кнопкой закрытия страницы, значительно улучшены средства обработки вкладок, плюс к этому появилась функция, позволяющая восстанавливать все открытые на момент выхода из программы страницы при следующем старте. Другое важное нововведение – антифишинговая система, позволяющая предотвратить заманивание пользователя на фальшивые web-сайты, создаваемые злоумышленниками с целью завладения конфиденциальными данными жертвы. В области удобства работы с браузером также произошли ощутимые изменения: помимо дальнейшего улучшения интерфейса, авторы впервые внедрили в программу средства проверки орфографии. Одновременно получили дальнейшее развитие средства поиска и встроенный клиент обработки RSS-новостей. Новая версия *Firefox* доступна для всех популярных платформ – Microsoft Windows, Apple Mac OS X и Linux, а также обладает тридцатью локализованными версиями интерфейса (в том числе, русским).

» Читайте обзор *Firefox 2.0* и взгляд на него изнутри Mozilla Foundation в LXF86!

www.mozilla.com



Linux Land 2006

Вторая по счету экспозиция LinuxLand (www.linuxland.ru) проходила в этом году в Москве, в стенах 69 павильона ВВЦ, в рамках выставки информационных технологий SofTool'06 (www.softool.ru) в период с 26 по 29 сентября. Ее организатором, как и в прошлый раз, выступала компания **LinuxCenter.Ru** (www.linuxcenter.ru) при содействии корпорации **IBM** (платиновый спонсор LinuxLand), **Novell** (золотой спонсор), **R-Style** (серебряный спонсор), **Mandriva**, **Webalta** и **.masterhost**. Параллельно с экспозицией проходила и международная конференция «ИТО-2006: Технологии Linux», организованная WDU и МПГУ при участии Московского Бюро ЮНЕСКО. Информационную поддержку обоих мероприятий оказывал журнал Linux Format.

Основная задача LinuxLand – собрать в одном секторе выставочной площади все известные Linux-компании – была решена на отлично. Посетители LinuxLand имели уникальную возможность задать свои вопросы непосредственно представителям IBM, Novell, Mandriva, Bitrix, R-Style, webalta, Etersoft (разработчик WINE@Etersoft), Smart Software, «Линукс-Онлайн» (дистрибутив Linux XP), НПО «Сеть» (MOPS Linux), ASPLinux, не говоря уже о LinuxCenter.Ru и Linux Format. Немного огорчало лишь отсутствие ALT Linux, которое, впрочем, было чисто символическим – собственный стенд компании был расположен в нескольких десятках метров от экспозиции. Хочется особо отметить, что на на многих стендах присутствовали не только менеджеры по связям с общественностью, но и инженеры-разработчики, а поскольку процент «хакеров» среди посетителей Linux Land был весьма высок, вокруг многих стендов быстро образовывались своего рода «тусовки по интересам». Достаточно ярким в этом смысле был стенд Novell, который еще раз доказал: трехмерный рабочий стол – это круто! Пока одни посетители с восхищением смотрели на дрожащие окна и капельки дождя в SUSE Linux Enterprise Desktop, другие толпились возле широкоформатного экрана, на котором мирно сосуществовали Nautilus и Проводник Windows. Xen – это тоже круто, а имея на руках два последних номера Linux Format, вы вполне можете проделать все то же самое (и даже больше) у себя дома. Да здравствует Open Source!

Не меньшим вниманием пользовался и стенд Mandriva, и не удивительно – ведь на нем можно было не только вживую пообщаться с вице-президентом Mandriva по продажам Полем Гули (Paul Guillett), но и увидеть в действии новейшую Mandriva Linux 2007, которая в тот момент находилась в стадии финальной «обкатки». Linux Land был вообще богат на премьеры от Mandriva – подробности ниже.

Доклады и мастер-классы

Как и любая уважающая себя выставка Open Source, Linux Land – это не только стенды и представительства. В течение всего времени работы экспозиции проходили презентации и мастер-классы, посвященные актуальным проблемам Open Source, применению его в бизнесе и т.п. Доклады были подобраны, что называется, на любой вкус, поэтому стулья в зале никогда не пустовали. Еще бы: – сложно пройти мимо такого заголовка, как «Платформа 1С: Предприятие на рабочих местах Linux» (Etersoft) или «Презентация новых серверов от HP пятого поколения» (R-Style) со ставшим уже традиционным выкатыванием в зал настоящего сервера HP ProLiant. Однако, гвоздем программы стала мировая премьера корпоративных решений Mandriva: Corporate Server и Corporate Desktop 4.0. Эти продукты были выпущены всего за несколько дней до старта Linux Land, и хотя корпоративная линейка Mandriva не столь популярна в нашей стране, как enterprise-решения Red Hat и Novell/SUSE, они привлекли к себе самое пристальное внимание слушателей. Неизменным интересом пользовалась и презентация «MOPS Linux 5.0 – универсальный дистрибутив для изучения технологий Linux». Кстати, по итогам прошлогодней выставки SofTool, MOPS Linux была присуждена престижная награда «Дебют года». Интересно, что принесет ему 2006 год?

Одновременно с презентациями на специально оборудованном стенде D16 проходили тренинги и мастер-классы. Пользуясь случаем, хочется сказать слова благодарности в адрес фирмы DESTEN Computers – это единственная компания, которая оказалась способна поставить 25 компьютеров для слушателей в самые сжатые сроки. Темы мастер-классов варьировались от таких продвинутых, как использование IBM WebSphere Application Server Community Edition (WAS CE), мощного открытого сервера приложений Java Enterprise Edition (тренер: Александр Цымбал, IBM), и обзор новейших возможностей SUSE Linux Enterprise 10, до самых базовых – например, знакомства с Linux. Таким образом, практически каждый желающий, независимо от уровня подготовки и области интересов, мог найти для себя что-то полезное. По стечению обстоятельств, стенд нашего журнала находился как раз напротив D16, что, конечно, было несколько утомительно (отвечать на вопрос: «Где проходят мастер-классы Linux Land?» каждые десять минут – это вам не шутки), однако мы могли воочию убедиться, что эти самые мастер-классы пользовались спросом со стороны посетителей экспозиции во все дни ее работы.

ИТО-2006: Технологии Linux

Очередная конференция «Информационные технологии в образовании» (ИТО-2006), традиционно проводимая Всемирным Распределенным Университетом (WDU) и Московским Педагогическим Госуниверситетом (МПГУ) при поддержке Центра компетенции Linux (IBM), в этот раз проходила «под одной крышей» с экспозицией Linux Land. В работе конференции приняли участие представители ведущих вузов и других образовательных учреждений России, а также бывших союзных республик: Казахстана, Таджикистана и Украины. Открытие конференции состоялось 26 сентября в Большом зале 69-го павильона ВВЦ: с приветствием к участникам выступили IBM, Novell, WDU, МПГУ, LinuxCenter.Ru и Linux Format.

Центральные доклады ИТО-2006 были посвящены опыту внедрения открытых решений и образователь-

ных программ на основе Open Source в России и странах ближнего зарубежья. Представители Казахстана рассказали о государственной программе всеобщего обучения компьютерной грамотности, которая будет реализована для 2 млн. граждан этой страны в 2007-2009 годах. Можно ожидать, что в основу данной инициативы ляжет Linux и другие свободные технологии.

Помимо слушания докладов, для участников конференции проводились специализированные мастер-классы, в ходе которых был представлен загружаемый дистрибутив «ИТО-2006: Linux / Open Source», базирующийся на SUSE Linux. Дистрибутив был разработан специалистами компании LinuxCenter.Ru с учетом требований государственной программы по информатике. Помимо традиционных компонентов – web-браузера и офисного пакета, в его состав были включены средства для разработки на языках Basic, Pascal и Prolog, не говоря уже о C/C++. Копия диска была бесплатно вручена всем зарегистрированным участникам конференции.

В ходе тренингов, участники конференции получили возможность освоить практическую работу в Linux/KDE и офисном пакете OpenOffice.org под руководством опытных инструкторов. Те, кто успешно прошел последовавшее за обучением квалификационное испытание (таких людей оказалось немало – 46), получили сертификат «Пользователь Linux»/«Пользователь OpenOffice.org», подписанный WDU.

Торжественное подведение итогов конкурса

IBM WebSphere Community Edition Contest 2006

Торжественное подведение итогов конкурса IBM WebSphere Community Edition Contest 2006, проводимого компанией LinuxCenter.Ru при поддержке корпорации IBM и информационной поддержке журнала Linux Format состоялось в первый день работы Linux Land – сразу же по завершении мастер-класса IBM WebSphere Application Server Community Edition (WAS CE), так что слушатели, только что освоившие работу с этим сложным инструментом, получили редкую возможность вживую увидеть и услушать людей, которые успешно используют его в своей профессиональной деятельности. Напомним, конкурс проходил в период с мая по сентябрь 2006 года, а его участникам предлагалось разработать любой продукт, логически связанный с WAS CE: готовое приложение, инструментарий, учебный курс, подробную и грамотную статью, и т.п. Практика показала, что интересы участников конкурса сфокусировались в трех основных номинациях: решения, инструментарий и методическое/информационное обеспечение, поэтому, посовещавшись, Экспертная комиссия решила выбирать победителей именно в них.



› **Александр Бабаев только что получил Гран-при – ноутбук Lenovo.**

Итак, внимание, наступает торжественный момент! Мы называем имена лауреатов конкурса IBM WebSphere Community Edition Contest 2006:

» В номинации **Решения** победителем был выбран Александр Бабаев и его мультиблог-система jDnevnik (www.jdnevnik.com). Александр стал обладателем суперприза – ноутбука Lenovo Thinkpad;

» В номинации **Инструментарий** победил Андрей Гребнев с проектом AtLeap (atleap.dev.java.net). Андрей получил от организаторов конкурса мультимедиаплеер Apple iPod Video;

» В номинации **Методическое и информационное обеспечение** лауреатом стал Иван Орехов, сотрудник компании StarSoft Development Labs, представивший программу курса «J2EE Development». Его приз – коробочная версия SUSE Linux 10.1 от компании LinuxCenter.Ru.

Помимо этого, Экспертная комиссия решила отметить специальной премией Антона Черноусова за его статью «Подход к реализации Web-приложения». Эта статья будет опубликована в одном из ближайших номеров Linux Format.

Помимо этого, все лауреаты конкурса получили специальные призы – годовую подписку на журнал Linux Format и подарочный сертификат стоимостью 10 000 рублей на любые товары из ассортимента интернет-магазина LinuxCenter.Ru.

Мы от души поздравляем победителей конкурса и спешим сообщить – это не последнее IT-соревнование, проходящее при поддержке нашего журнала! Внимательно следите за анонсами на www.linuxformat.ru и крупных новостных сайтах – вас ожидает еще много нового и интересного! LXF

J2EE Development

В рамках учебного курса «J2EE Development» изучаются основы разработки enterprise-приложений с помощью известных открытых технологий, таких как Struts и Hibernate. Процесс обучения построен на примере создания классического enterprise-приложения – системы обработки заказов, но обладающей лишь базовой функциональностью. Сначала студенты знакомятся с основными технологиями J2EE – Java Servlets и Java Server Pages и, используя для доступа к данным JDBC, разрабатывают исходную версию приложения. Затем они выполняют миграцию на технологии Open Source, заменяя Java Servlets и JSP-компоненты на Struts и Tiles, а JDBC – на Hibernate. В качестве сервера J2EE приложений используется IBM WebSphere Application Server Community Edition, а в качестве СУБД – MySQL. Данный курс читается специалистами StarSoft Development Labs (http://www.starsoftlabs.ru) и рассчитан на один месяц. Обучение бесплатное, а слушателям выплачивается стипендия.

jDnevnik

» **Web** jdnevnik.com » **Лицензия** CDDL

jDnevnik – свободно распространяемая блог-система. С помощью jDnevnik можно вести собственный интернет-дневник или организовать хранилище фотографий. Одной из отличительных особенностей jDnevnik является наличие всех необходимых инструментов для ведения корпоративного блога – в частности, мощной иерархической системы прав. Доступ регулируется на уровне отдельных объектов, кроме того, существует система лицензий, позволяющая быстро изменить права того или иного пользователя. В jDnevnik может существовать более миллиарда миллиардов различных объектов: блогов, записей, комментариев, пользователей и категорий.

AtLeap

» **Web** https://atleap.dev.java.net » **Лицензия** Apache 2.0

AtLeap – это Java-каркас для быстрого построения на его основе web-приложений. AtLeap инкапсулирует около 50 различных библиотек. Он протестирован на десяти серверах приложений (в том числе, на IBM WebSphere Community Edition 1.0.1.2), семи СУБД, трех виртуальных машинах Java, шести браузерах. AtLeap включает многоязычную CMS с полнотекстовым поиском на 13 языках. Система может индексировать файлы в форматах RTF, PDF, Word, Excel, PowerPoint, HTML, XHTML и других. Консоль управления переведена на 6 языков. AtLeap использует два DHTML WYSIWYG-редактора и содержит более 5 Мб исходного кода.

РЕПОРТАЖ С МЕСТА СОБЫТИЙ

LinuxWorld

Какая технология произведет главный фурор в этом году? **Ник Вейч** истоптал ковры в Moscone, пытаясь разобраться.

Вслед за праздником хакеров OSCon прошла более крупная LinuxWorld Conference & Expo: Сан-Франциско (14–17 августа 2006 года), главное событие в календаре Linux после того, как нью-йоркское шоу пришло в упадок. Если вы придете в Moscone Center в надежде поглядеть на пингвинов, то вас, очевидно, постигнет разочарование – хотя сообществу и технологии уделяется большое внимание, упор сделан на функционирование и внедрение Linux в бизнесе.

Но все же на территории .org в этом году определенно царило праздничное настроение. Уютная гостиная Slashdot полнилась усталыми путниками, а прилегающие ячейки сообществ порою казались тихими гаванями в бурном корпоративном море. Здесь, среди прочих, можно было обнаружить Creative Commons (приятно видеть вас снова); проект LTSP; вторжение FreeBSD; Gnome и KDE (рядышком!) и Fedora. Мы немного постояли у стенда Fedora, и вопрос номер два (после «А как сделать, чтобы Fedora Core 5 воспроизводила MP3?») был «А где стенд Red Hat?» Хороший вопрос: таковой вообще отсутствовал! Весьма странное упущение, особенно при том, что новый дружок-соперник Red Hat – Novell – разместил огромный стенд прямо у входа, и все его презентации и демонстрации были явно нацелены на корпоративный рынок.

Нат Фридмен [Nat Friedman] из Novell весьма впечатляюще продемонстрировал новый настольный SUSE Linux Enterprise Desktop (SLED) для предприятий, как на стенде, так и в ходе конференции. SLED 10 приняли тепло, а во время презентации ходили разговоры о том, что Lenovo, недавно перекупившая бизнес ThinkPad от IBM, будет выпускать линейку ноутбуков с установленным на них SLED. Novell ожидали приятные новости – SLED наградили как Лучший Продукт Выставки (Best Product At Show) и Лучший Настольный Продукт (Best Desktop Product). Выходит, он делает то, что надо.

> «Один изготовитель берется установить Linux где угодно: от самолетного кресла до бензонасоса.»

SUSE и Fedora не были единственными дистрибутивами, представленными на экспозиции. Наряду с главным стендом Ubuntu представил свой стенд в зоне .org, и его хозяева не успевали раздавать бесплатные диски – но мы-то знаем, многие подходили просто чтобы взглянуть на некий замечательный журнал про Linux с Ubuntu на обложке, тоже выложенный на стенде…

«Железо»

Для фанатов «железа» на выставке было немало интересных стендов – например, некоторые изготовители встроенного оборудования предлагали установить Linux в совершенно невообразимых местах: от самолетных кресел до бензонасосов .

Что касается хранения данных, на выставке красовались новинки от 3ware и Promise. Похоже, что Promise в данный момент является лидером технологии благодаря SATA RAID с новыми скоростными устройствами, которые проходят тестирование. Promise также не сомневается, что вскоре сможет предложить сильный SAS (Serial Attached SCSI). Мы надеемся, что удаст-



Фото с LinuxWorld: David Blankenhorn, sjphotography.com

› По сравнению с прошлым годом, корпоративная тема урвала больше внимания, но и защитники свободного ПО не сдали позиций.







ся получить оборудование для тестирования – даже с SATA RAID, технология так отстает от теоретической производительности, что становится непонятным, ради чего все это. Изготовители дисков говорят, что скорость работы устройств в ближайшем будущем не сравняется с теоретической.

Мобилизация

Однако хакеры не отличаются постоянством, и после первого дня выставки разговоры опять перешли исключительно на мобильную технологию. Благодаря предложенной Nokia схеме проката, многие из тех, кто посещал стенды, были экипированы планшетами N770. В первый же вечер Trolltech объявил о выпуске Qtopia Greenphone. Подчеркивая, что фирма не собирается выходить на рынок телефонии, она во время выставки на самом деле запустила целый пакет (включая и телефон), чтобы подвигнуть разработчиков Qtopia на создание еще большего количества программ для мобильных устройств. С точки зрения открытого кода, телефоны – крепкий орешек: низкоуровневое ПО для них – с закрытым кодом, и очень жестко регулируется во всем мире.

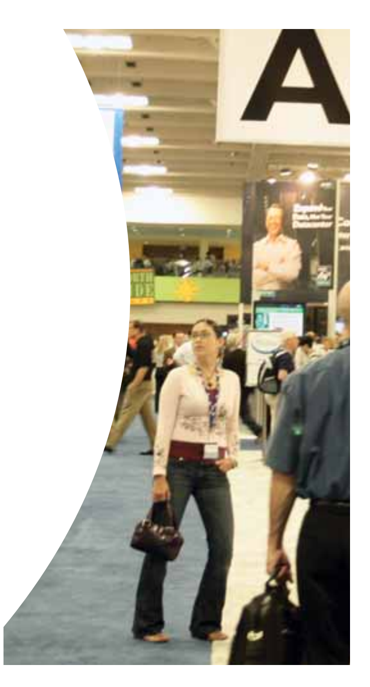
Тему мобильной связи продолжал огромный стенд Motorola. Motorola не так давно открыла большинство своих программ и очень хочет, чтобы ее рассматривали как члена сообщества. Возможно, нам даже удастся увидеть еще больше мобильных телефонов ее производства на базе Linux за пределами Японии. Распространение Linux продолжается – раньше он добирался только до смартфонов, а сейчас уже рассматривается и в качестве технологии для обычных моделей.

В зале было еще много интересного, на что стоило взглянуть и что стоило сделать, однако те, кто не сумел добраться через туннель на другой конец Moscone Center, определенно пропустили отличные семинары. В этом году их было даже больше, чем обычно – особенно удались семинары OSDL, хотя нам и не удалось поприсутствовать на всех. IBM также спонсировала серию бизнес-семинаров и встреч, привлекших внимание многих посетителей.

Но интереснее всего были, наверное, не отличающиеся строгой организацией групповые дискуссии «по интересам», проходившие по вечерам. Хотя мероприятие продолжалось четыре дня (конференция началась за день до выставки), выбор того, на что стоило взглянуть и что стоило попробовать, оказался прямо-таки чересчур велик. Если бы вы каждый год могли участвовать только в одном из событий, посвященных Linux, то это было бы именно оно – итак, увидимся там же в августе 2007! LXF

Пишите нам

Вы были на выставках Open Source – возможно, на тех двух, о которых мы вам здесь рассказали? Поделитесь впечатлениями:
letters@linuxformat.ru





› Intel, HP, IBM. Почему все подлинно корпоративные стенды – голубые?



› Сколько дадите ? Интересно, а почём номерные знаки XP...

› Нат Фридмен объясняет, как найти рисунок кошки. Опять!




› Наконец-то! Телефон, на котором я могу работать в *Emacs*...



LINUX FORMAT

Обзоры



Новинки программного и аппаратного обеспечения в описании наших экспертов



Алексей Федорчук
Свою первую (и последнюю) программу написал еще на Алголе.

На злобу для или Oracle vs Red Hat

Недавно все сообщество Open Source всколыхнула новость о том, что Oracle будет выпускать свой Linux, предназначенный для запуска своей же СУБД. И не просто свой дистрибутив – а Red Hat цельнотянутый, освобожденный от «ненужных» компонентов вроде MySQL или PostgreSQL. Первая реакция, естественно, была: вот он, звериный оскал капитализма, бездушного и бездуховного,

> «У Oracle есть два пути...»

при котором все покупается и все продается. Однако, если отбросить эмоции и вдуматься, что уж такого страшного произошло?

Действительно, у Oracle есть два пути. Первый – делать свой дистрибутив узконишевого назначения – для запуска своей СУБД. Эта ниша хоть и глубока (в финансовом отношении), но все равно ниша, широкие массы трудящихся никак не затрагивающая. Второй же вариант – создавать инфраструктуру для поддержки своего детища, то есть, в конечном счете, вкладывать в Open Source силы и средства. А это, в итоге, новые рабочие места для программистов открытого ПО. От чего сообществу никакого вреда, окромя пользы, быть не может...

Ну а разговоры о морали, нравственности и тому подобных материях – они, конечно, интересны, но оставим их потомкам. Если ребята из Red Hat покажут себя настоящими мужиками и в этой драке выстоят – что ж, уважение сообщества им гарантировано. Если нет – вспомним слова Олега Куваева: «Тех, кто утонул, замерз, умер от голода, спился – их не было здесь. И даже память о них затерлась...»

alv@posix.ru

Сегодня мы рассматриваем...

Cairo с. 19

graham/build/cairo-demo/cairo-demo/kapow/kapow.png

LXF

› Масштабируемая, анимированная графика – готов ли Cairo API к промышленному применению?

Blender с. 16



› Первый обзор инструмента 3D-моделирования с момента выхода на экраны Elephants Dream.

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.



Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют *Autopackage*, мы следуем этому совету.



Вердикт

Google Earth
Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» *Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.*

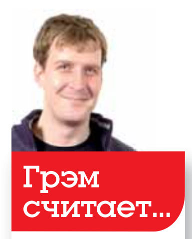
Рейтинг 9/10

Gnome 2.16

Даешь разнообразие – Gnome получил встроенный редактор меню! Уж теперь **Энди Хадсон** заварит кашу…

Вкратце...

» Тяжеловесная рабочая среда. ***См. также:*** KDE или другие, более легкие, оконные менеджеры (*Xfce*, *Fluxbox*).



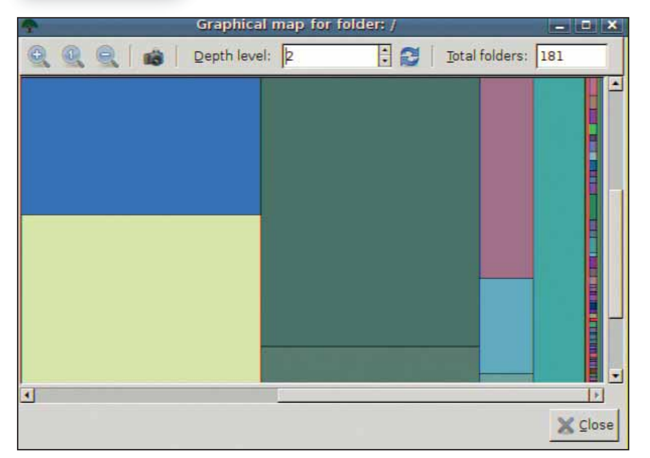
Грэм считает...

«Честно говоря, иногда пользователи Gnome меня просто поражают. *Tomboy* чем-то смахивает на Тома Круза, а разработчики Gnome напоминают секту сайентологов, распространяющих свою веру среди благодарного человечества. Засуньте Gnome в рождественский носок, а сами пользуйтесь KDE».

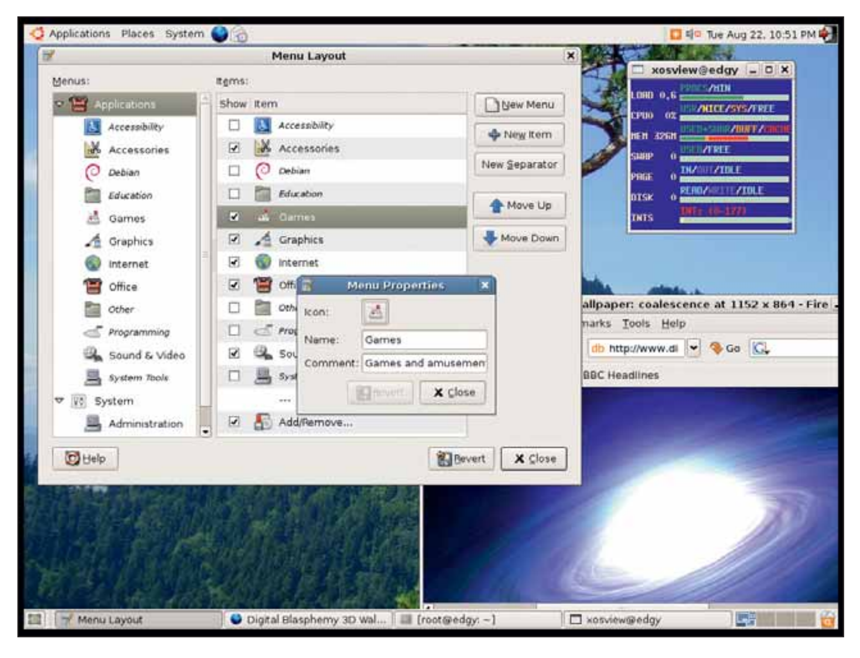
Название наводит на мысль о карликах, но Gnome растет и хорошеет с каждой новой версией. Два крупнейших коммерческих дистрибутива (SUSE и Red Hat) открыто поддерживают Gnome как среду рабочего стола, вместе с репутацией его положения «по умолчанию» в Ubuntu, кажется, сулит ему безоблачное будущее. Последняя версия содержит немало заметных новшеств, явно способных вызвать резонанс в среде открытого ПО.

Gnome 2.16 теперь поставляется с *Gtk#* и Mono, их больше не нужно загружать отдельно. Mono реально начинает пробиваться в стан крупных дистрибутивов, поэтому приятно видеть поддержку на уровне рабочей среды. Единственный недостаток видится в неминуемом увеличении объема – данная версия занимает на 5 МБ больше предыдущей: 120 МБ в виде архива. Планы Sun по открытию Java означают, что у Mono появился реальный конкурент по кросс-платформенной разработке, но при столь тесной увязке с Gnome Mono определенно получает фору. Первый признак включения Mono – *Tomboy*, приложение для экранных заметок. Нам очень понравился этот удивительно удобный инструмент, с которым можно не только быстро делать заметки, но и связывать их между собой в стиле wiki.

Приход Mono – хорошая новость также для людей со слабым зрением, так как в Gnome теперь появилась *Orca*, технология, применяющая комбинацию из «увеличительного стекла», перевода текста в речь и ввода по Брайлю. *Orca* вытеснила из Gnome программу *Gnopernicus* с похожей функциональностью, причем с ведома и благословения разработчиков *Gnopernicus*.



› ***Baobab*** **в стиле Пита Мондриана мигом покажет, которая из директорий слопала ценное дисковое пространство.**



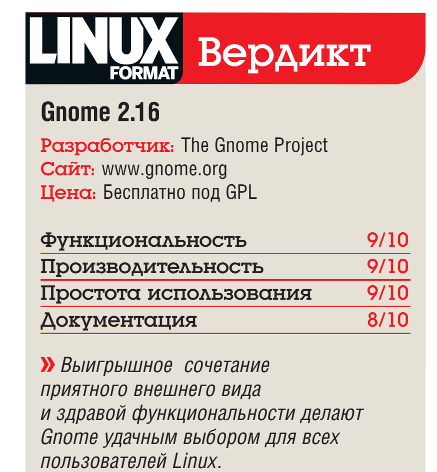
› **Приготовьте себе богато украшенное аппетитное меню с помощью *Alacarte*.**

Еще одна удачная интеграция – *Alacarte*, редактор меню, прежде известный под неудачным названием *Smeg*. С этим удобным редактором вы быстро настроите ваши меню «Программы» и «Система». Можно удалять приложения из меню (но они все еще будут доступны через командную строку), а можно и добавлять пункты к существующим меню и подменю. Совсем недавно привести меню в порядок было проблематично: по умолчанию Gnome такого не делал. *Alacarte* упрощает эту задачу, придерживаясь философии простоты Gnome.

A-gnome-стицизм

Работа по интеграции дистрибутиво-независимых административных инструментов продолжается, и в первые ряды шагнул *Baobab*, графический дисковый анализатор. Одна из экзотических функций *Baobab* – графическое представление размеров директорий (см. рис.). К этой россыпи мелких квадратиков надо попривыкнуть, прежде чем наступит ее осмысление, зато вы мгновенно засечете чересчур разросшийся файл.

В порядке критики можно сказать, что Gnome пренебрегает возможностью представить броскую версию типа «вот я какой», предпочитая неторопливое поступательное развитие. Пример – *Gnome Power Manager*: Ubuntu и Fedora включали его в свои пакеты и раньше, но только сейчас *GPM* официально вышел на рабочий стол. И все же Gnome еще никогда не был так хорош, за что спасибо Cairo 1.2 (включенному благодаря *GTK 2.10*). Графика четкая, со сглаженными краями и отделкой окон, которая так и прыгает с экрана. Ну, не могут пользователи Ubuntu, все как один, сделать неверный выбор! LXF



LINUX FORMAT Вердикт

Gnome 2.16

Разработчик: The Gnome Project
Сайт: www.gnome.org
Цена: Бесплатно под GPL

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Документация	8/10

» *Выигрышное сочетание приятного внешнего вида и здравой функциональности делают Gnome удачным выбором для всех пользователей Linux.*

Рейтинг 9/10

VMware Server 1.0

Виртуализация – это большие деньги, и лидеры программной индустрии бьются за покупателя. **Грэм Моррисон** дегустирует плоды войны: бесплатное ПО корпоративного уровня.

Вкратце...

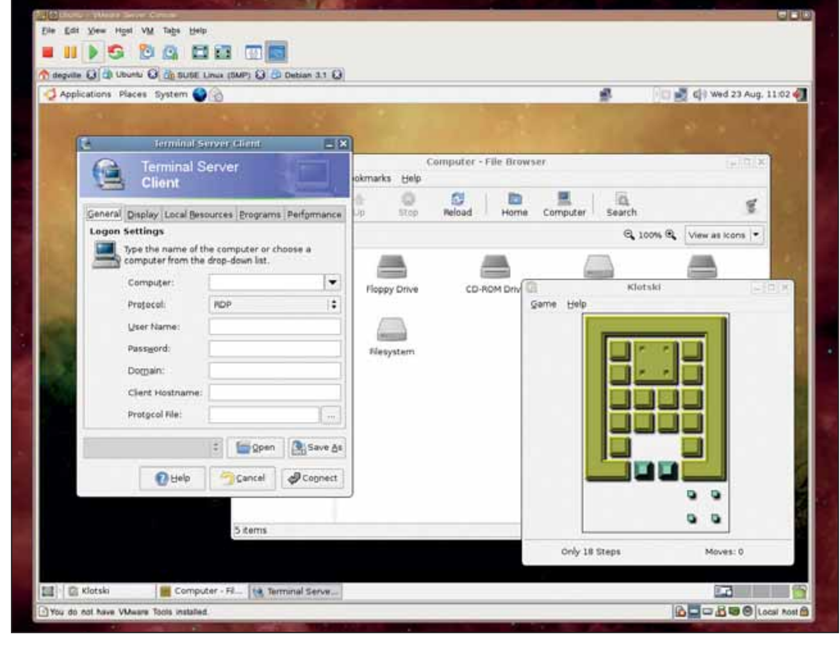
» Бесплатная версия популярного ПО для серверной виртуализации от VMware. ***См. также***: *Xen и Parallels Workstation.*

Виртуализация стала полем битвы. *VMware* и Microsoft воюют за лидерство, а Xen и Parallels Inc. – за то, что останется. К счастью, свирепая борьба привела к выпуску бесплатного ПО от обоих главных игроков и к обещанию бесплатных версий серверных технологий.

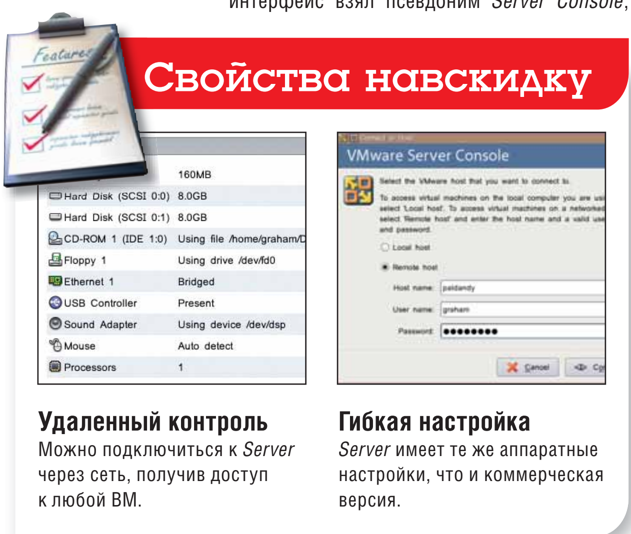
Идея серверной виртуализации проста. На одной машине можно запускать несколько изолированных серверов, каждый в своей виртуальной рабочей среде. Например, хостинговая компания содержит каждый из своих многочисленных сайтов на отдельном компьютере, для обеспечения управляемости. Это значит, что каждая машина использует только малую часть своих ресурсов, часто меньше 5%. Перемещение физических серверов в виртуальный домен позволит предприятию усилить аппаратный потенциал.

VMware Server – не что иное, как производное известного продукта *VMware*, нашумевшего *GSX Server*. В начале года его продавали за тысячи долларов. Но идея *VMware* состоит в том, что фирмы, опробовав бесплатную версию и осознав, как это здорово, непременно захотят приобрести коммерческую версию *GSX*.

Установить *Server* можно как из RPM, так и через сценарий оболочки; затем последует настроечный скрипт, который задаст вам с десяток вопросов о ваших системных требованиях. Они кажутся сложными, пока вы не поймете, что для большинства пунктов вполне сойдут ответы по умолчанию. Фактически, *Server* – это более-менее полнофункциональная версия *VMware Workstation*, настольной программы виртуализации от *VMware* (см. обзор в **LXF75**). Основной графический интерфейс взял псевдоним *Server Console*, но интерфейс *Workstation* здесь копируется чуть ли не с точностью до пикселей. Здесь есть мастер создания новых машин, панель настройки для редактирования виртуальных устройств и вкладки с консольными окнами для переключения между машинами. Вы можете создавать новые машины, открывать существующие и запускать их совместно точно так же, как с *Workstation*, при той же поддержке для многопроцессорных машин, включая 64-битные (если таковые имеются). Созданные виртуальные машины (ВМ) можно использовать как с *VMware Player*, так и с *Workstation*, и мы не встретили проблем с запуском коллекции из Debian, Ubuntu и Windows XP. Можно даже открывать ВМ, созданные *Virtual PC* от Microsoft.



› *VMware Server* – **вылитый Workstation, только без множественных снимков.**



Свойства навскидку

Удаленный контроль
Можно подключиться к *Server* через сеть, получив доступ к любой ВМ.

Гибкая настройка
Server имеет те же аппаратные настройки, что и коммерческая версия.

Удаленный обмен

Единственная функция, отличающая *Server* от *Workstation* – отсутствие множественных моментальных снимков системы (snapshot), позволяющих отменять изменения и отслеживать по порядку все инсталляции на ваших ВМ. *Server* ограничивается одним снимком – но это больше, чем *Player*. На удивление, *Server* имеет одно важное преимущество перед *Workstation*: вы можете удаленно соединяться с ним при помощи второй копии *Server*. Виртуальные машины, запущенные на сетевом компьютере, доступны не просто интерактивно, а прямо во время работы. Это открывает двери всем видам удаленного администрирования и кооперативной разработки. Потерялся только звук.

По сравнению с бета-версией производительность возросла неизмеримо. Пока память позволяет, виртуальные машины крутятся почти с той же скоростью, что в штатных инсталляциях – поразительная мощь задаром! Важно только не забывать различия между «свободой слова» и «бесплатным пивом». Если вы подумываете о виртуализации своих серверов, будьте осторожны – *VMware* запросто может прекратить поддержку бесплатной версии, оставив вас сироткой без всякой лицензии. LXF

LINUX FORMAT Вердикт

VMware Server 1.0

Разработчик: VMware Inc
Сайт: www.vmware.com
Цена: Бесплатно по закрытой лицензии

Функциональность	8/10
Производительность	8/10
Простота использования	7/10
Документация	7/10

» *Server выглядит и действует так же, как коммерческий продукт. Приобретение хорошее, но будьте осторожны, доверяя ему свое будущее.*

Рейтинг 8/10

Blender 2.42a



Стоит ли рассматривать версию, обновленную лишь на 0.01? Конечно, стоит, если речь идет о *Blender*. **Бен Харлинг** объясняет, почему.

Вкратце...

» Моделирует, анимирует и отрисовывает 3D-графику, редактирует видео и т.п. Солидная свободная альтернатива *3D Max* или *Maya*, со всеми удовольствиями.

Один из бесспорных лидеров свободного ПО, *Blender 3D* (ныне известный как просто *Blender*), со времени своего создания в качестве внутреннего 3D-инструмента фирмы NaN Technologies, претерпел несколько гигантских обновлений. Когда фирма развалилась, его исходный код был приобретен на общие средства своего уже вполне состоятельного пользовательского сообщества и опубликован. Результат здорово превзошел ожидания, и сегодня программа являет собой реальную альтернативу «полноценным» 3D-приложениям.

Итак, что же нового? Немало. Наберется на несколько новых версий, но в соответствии с консервативной манерой нумерации *Blender*-сообщества, официально номер увеличился только с 2.41 до 2.42. Многие новые функции появились благодаря недавнему проекту Orange, среди целей которого было не только

> «Рендеринг в Linux идет заметно быстрее, чем в Windows и OS X.»

создание анимированного фильма, но и придание *Blender* новых возможностей по производству фильмов.

Первая и главная из них – новый дизайнер/конструктор материалов на базе «узлов». Пользователи *Maya* и *Cinema 4D* знакомы с концепцией «связываемого» ввода, фильтров и иконок для визуального отображения материала – давняя мечта пользователей *Blender*.



› Может, это и не лучшая в мире короткометражка, но *Elephants Dream* (показан в Blender) впечатляет технически, отлично смотрится, и стал весомым вкладом в развитие проекта.

Поразительно чистый и эффективный редактор узлов подарил пользователям бесконечную свободу визуального творчества; он обеспечивает надежный предпросмотр в реальном времени, который можно настроить, добавив собственную сетку. Чего в нем нет (как и вообще в *Blender*'е) – это простоты использования, но глубины его возможностей хватит на горы диссертаций.

Blender продвинулся еще на шаг вперед в использовании системы узлов, применив ее также для управления работой рендерера. Эта функция прекрасно реализована и наделяет художника полной властью над любым объектом рендеринга. Нужен хроматический ключ? Нет проблем! Несложная настройка узлов в *Blender*, и вы готовы строить «живой эфир» вашей сцены на голубом экране, не привлекая другие приложения – а в коммерческих программах без этого не обойтись.

Крутой рендеринг

Движок рендеринга *Blender* полностью обновлен, наделен функцией поточной обработки и автоматического объединения отдельных слоев рендеринга. Компоненты рендеринга аккуратно внедрены во все остальные части приложения, что дает возможность пользователю встраивать окна рендеринга в стандартные 3D-сцены и оперативно редактировать предпросмотр. Стандартный для *Blender* построчный рендеринг здорово усовершенствован, и по качеству работы вплотную подошел к уровню, доступному трассировке лучей.

Если ваша цель – фотореализм, обратитесь к рендереру *Yafary*, который теперь полностью интегрирован в пользовательский интерфейс и производит изображения потрясающей точности, правда, ценой ощутимого увеличения времени рендеринга. Есть скрипты экспорта для других трассировщиков лучей, например, *Sunflow*, превосходной системы Indigo и даже *Mental Ray* (стандарт де-факто в данной индустрии). Разработчики также доработали алгоритм встроенной трассировки лучей для более достоверного отображения стекла и других отражающих поверхностей, включая достоверную прозрачность и внутреннее отражение. Что особенно радует, Linux по времени рендеринга в *Blender* заметно опережает Windows и OS X.

Красота модели

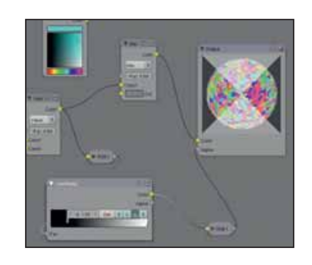
Как и следовало ожидать, налицо множество захватывающих новшеств по части моделирования (несомненно, это одна из сильных сторон *Blender*), не снижающих прежнюю, одну из быстрейших, скорость создания сеток. Есть новые органы управления оснасткой персонажей, включая слои каркасов. Это равносильно нескольким каркасам, влияющим на одну и ту же сетку, которые можно включать и отключать в соответствии с потребностями конкретной сцены.



Свойства навскидку



Моделирование жидкостей
Улучшенный симулятор жидкостей невероятно реалистично брызгается.



Узелковый материализм
Видеокомпозиция и создание материалов упрощаются с помощью настройки узлов.

Что еще новенького?

» UV-развертка Не новая, но значительно усовершенствованная функция, с упрощенным подходом к маркировке швов и автоматической разверткой по направляющим.

» Python API Версия 2.42 включает усовершенствования для языка сценариев BPython, включая PyDrivers – вывод анимации с помощью выражений Python.

» Библиотеки Использование элементов из других файлов путем связывания и встраивания библиотек. Обновлено с целью повышения стабильности и добавления новых функций.

» Скрипты Python Существуют сотни сторонних скриптов BPython, покрывающих все нужды, от изображения пчелиного роя до обеспечения кросс-платформной совместимости производимых игр (буквально один щелчок для Linux, Windows или Mac).

» Редактор видеоряда Интеграция библиотеки *FFMPEG* (эксклюзив Linux) обеспечит гораздо большую отзывчивость программы и, теоретически, работу с видеоматериалами в реальном времени.

» А еще... Мы коснулись лишь немногих из примерно 3000 изменений и дополнений, появившихся со времени выхода прежней версии.

Все эти усовершенствования наделяют *Blender* наиболее развитыми функциями отделки персонажей среди всех 3D-пакетов – добрая весть для анимационных студий, решивших использовать его в работе. Работа с персонажем традиционно была пробным камнем для распознавания профессионала, в отличие от любителя, просто нахватавшегося верхов. Большинство пользователей *Blender* попадают во второй лагерь. Одна из причин медленности распространения *Blender* как профессионального инструмента – нехватка ресурсов поддержки, которыми обладают знаменитые пакеты: примеров, учебников, помощи пользователям и т.д.

Но *Blender* не сдается, предлагая все новые усовершенствования своих и так авангардных на сегодняшний день инструментов моделирования. Новая функция группировки объектов позволяет управлять большими объемами сеток и анимационных данных быстро и удобно, независимо от привычного принципа наследования. Стек модификаторов подвергнут заметным обновлениям, включая мощный модификатор массивов как превосходную альтернативу устаревшей «дубликатной» системе прежних версий. Жалко, что модификаторы *Blender* пока не вполне нелинейны, а иногда норовят «заморозить» сетку, не дождавшись завершения редактирования.

Улучшились дела с чудесной «жидкостной» динамической системой, остающейся одной из лучших в этой области. Благодаря обновлению до 2.42, пользователь может теперь добавлять к жидкостным средам движущиеся объекты, струи и препятствия и любоваться вполне достоверными всплесками.



Ник считает...

«Проделан феноменальный объем работы. Думаю, поддержка Python значительно упростит интеграцию с другими инструментами.»

Крупный калибр

Результаты напряженной работы трудяг-эльфов *Blender* продолжают удивлять. Его встроенный игровой движок существенно усовершенствован и теперь справляется с настоящими шейдерами, позволяя программировать сложные визуальные эффекты, такие как HDRI (high dynamic range imaging). Движок физики *Bullet*, признанный стандарт, одинаково пригоден как для применения в реальном времени, так и для физического моделирования, которое можно зафиксировать для последующей вставки в полностью готовые сцены. В общем, производительность *Bullet* весьма прилична, с задачами он справляется либо с минимальными задержками, либо вообще без них.

Интеграция нового движка пока не завершена, и это означает частичную несовместимость со старым движком, Sumo. К тому же, встроенный скриптовый интерфейс, хотя и развивается стремительно, быстро отказывает при попытке полноценного управления игровой механикой. Если рассматривать его как часть общего пакета, игровому движку *Blender* как-то не хватает цельности и завершенности, присущих программе в целом.

Оборотная сторона медали

Взявшись за *Blender* , многие бросают его, не выдержав и нескольких минут, ибо чрезмерно сложный интерфейс напрочь «сдвигает крышу» новичкам. Плохая новость для них (или хорошая – для прочих): здесь ничего не изменилась. *Blender* покоряется только тем, кто «на ты» с пользовательским интерфейсом (особенно с горячими клавишами). Освоите их – работайте в удовольствие, а нет – только и будете рвать на себе волосы. Но заметьте: одолев замысловатый интерфейс, вы запросто перекроите его по своему вкусу, а различные экранные раскладки можно сохранять в файле для быстрого переключения между разными стилями редактирования.

Мучений можно бы избежать, если бы пакет включал набор учебных ресурсов, но, к несчастью, в деле изучения программы вы предоставлены сами себе. Полезны wiki и web-сайт *Blender* , но лучше поискать информацию на сторонних сайтах и форумах, хотя это трудоемко и не всегда продуктивно. К тому же среднестатистический пользователь *Blender* не богат яркими, вдохновляющими работами, на которых можно поучиться. Стыдно, при столь широких возможностях! Яркий пример – имитация жидкостей: опытный пользователь настроит ее за секунды, а новичку для этого понадобятся годы клавиатурных мытарств. Может быть, когда опытные художники повернутся лицом к программе, дела пойдут иначе, ну а пока львиная доля ее потенциала скрыта в тумане.

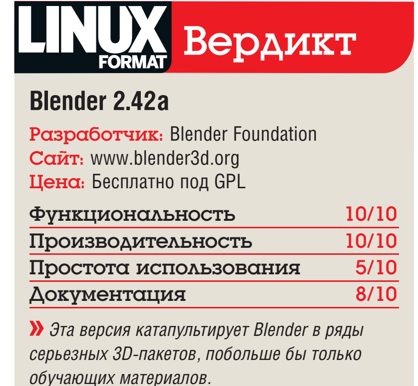
На горизонте

Мы не беремся предсказать, когда (если) разработчики пересмотрят интерфейс, но от темпов разработки *Blender* захватывает дух (предыдущая версия вышла меньше чем семь месяцев назад), и прогресс отнюдь не прекратился. Ходят слухи, что следующая версия поднимет программу на уровень 2.5, и если экстраполировать по различию между 2.41 и 2.42, новации должны быть поистине ошеломляющими. Один из проектов, который уже на конвейере – обновление несовершенного игрового движка с прицелом на модульный принцип работы. Пользователь должен получить свободу выбора между физическим и игровым движками в условиях преемственности проектов.

У сообщества пользователей *Blender* слюнки потекут при вести о возможной интеграции фантастического движка *Ogre 3* как основного рендерера интерактивного вывода *Blender* (опытные сборки по этой технологии уже появились). Интеграция *Ogre*, несомненно, выведет *Blender* в первые ряды интерактивного 3D-дизайна, соединив простую в обращении архитектуру создания игр с самой мощным на сегодняшний день движком рендеринга реального времени – как не закружиться головам адептов *Blender* ? LXF



» Система узлов дает пользователю полную власть над визуальным выводом и устраняет нужду в сторонних постпроцессорах.



LINUX FORMAT Вердикт

Blender 2.42a

Разработчик: Blender Foundation
Сайт: www.blender3d.org
Цена: Бесплатно под GPL

Функциональность	10/10
Производительность	10/10
Простота использования	5/10
Документация	8/10

» Эта версия катапультирует Blender в ряды серьезных 3D-пакетов, побольше бы только обучающих материалов.

Рейтинг 9/10

Xandros Desktop 4

Vista запаздывает, и продавцы настольных Linux-систем чуют запах крови… и денег. **Энди Ченнелл** пробует альтернативу для беженцев с Windows.

LINUX FORMAT TOP STUFF

Вкратце...

» Нацеленный на домашнего пользователя дистрибутив на основе Debian. *См. также* Ubuntu, Linspire и Mandriva.



Пол считает...

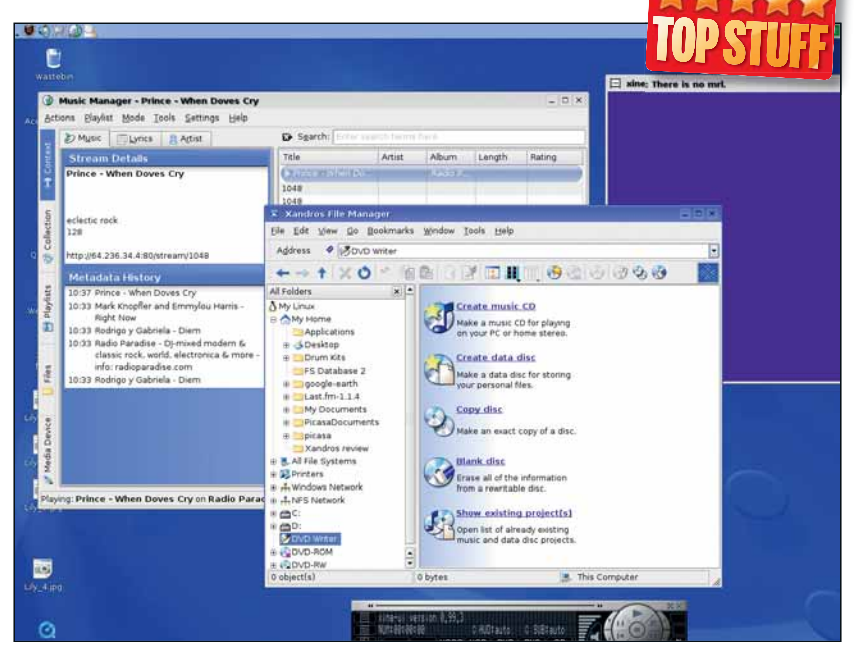
«Очень удобный дистрибутив, но какой ценой? По отчужденности от идей свободного ПО его может переплюнуть разве что сборка Linux от MS. Вам придется пожертвовать еще $79 в пользу FSF, чтобы хоть как-то загладить свою вину».

Xandros, восставший из пепла сгоревшего Corel Linux, всегда шел в авангарде Windows-подобных дистрибутивов. Его новейшая коробочная версия, Xandros Desktop 4, продолжает эту традицию. В XD4 есть программа *Progression Desktop* от Versora, которая собирает персональные данные пользователя – включая настройки Web, учетные записи почты, журнал, параметры рабочего стола и др. – в Windows-разделе и помещает их в Linux-раздел, превращая Xandros в зеркало Windows-системы.

Мигранты с Windows могут прихватить с собой программы Win32, включая *iTunes*, *Quicktime*, *MS Office*, *Photoshop* и др., благодаря программе *CrossOver Office Professional*. *CrossOver* – действительно замечательный продукт, удачно встроенный в рабочий стол Xandros. Интерфейс программы позволяет инсталлировать и запускать Windows-приложения через меню KDE так, словно они и вправду работают в Windows, без всякой возни с перезагрузкой. Более того, на зависть другим дистрибутивам, Xandros умеет читать и записывать данные на NTFS-разделах.

Свежее – лучше

У нас были проблемы с установкой прежних версий Xandros поверх других дистрибутивов (мы пытались сохранить раздел **/home**); с теми же трудностями мы столкнулись и на сей раз. Если вы обновляете существующий Xandros, все проходит прекрасно, но пользователям других дистрибутивов лучше создать резервную копию **/home**, а инсталляцию провести заново. Сама инсталляция проходит быстро и просто: система, готовая к web-серфингу, ставится чуть более 15 минут с минимальным пользовательским вмешательством, при этом все аппаратные части настраиваются корректно.



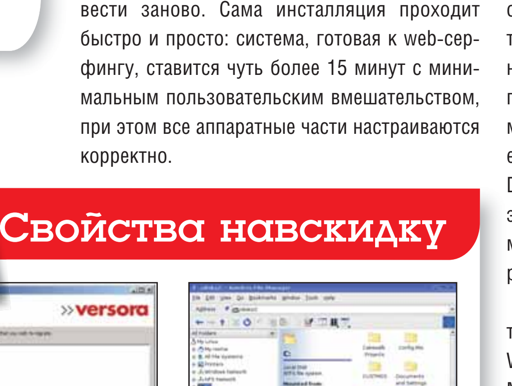
› XD4 имеет полный пакет медиа-программ, включая (с некоторыми нюансами) воспроизведение DVD.

Xandros придерживается KDE 3.4, предпочитая стабильность броской внешности, и знакомый мастер первоначальной настройки проводит обычную работу по сортировке локальных настроек, назначению кнопок мыши и т.п. Рабочий стол и структура меню хорошо организованы, и по умолчанию устанавливается «правильный» набор приложений. Пакетами распоряжается превосходный клиент *Xandros Networks* (*XN*). *XN* можно настроить так, чтобы он не выходил за границы собственных пакетов Xandros (их тысячи, и все они помещены на второй CD для пользователей без широкополосного доступа), а можно одним щелчком мыши перейти в хранилище неподдерживаемых пакетов Debian. «Не-Xandros» пакеты Debian инсталлировать тоже можно, хотя при этом возрастает вероятность сбоев. Тем не менее мы установили множество программ из различных источников без единой запинки.

Что касается внешности и поведения, то XD4 отлажен и стабилен, и пользователь Windows почувствует себя здесь как дома. Может быть, рабочий стол не самый крутой – некоторой магии *Xgl/Beagle* ему недостает – зато он справляется со своей главной задачей: быть и понятным новичку, и достаточно мощным для «зубра». Миграционные инструменты работают прекрасно – хотя Versora мало чем может помочь пользователям web-почты – и основная функциональность (по части работы в Сети) ложится на одно из расширений *Firefox*, Browser Sync от Google.

Есть много причин рекомендовать этот дистрибутив для домашнего использования: превосходная документация, хорошая работа с мультимедиа, беспроблемный доступ к Flash, Java и *Acrobat*, удобная запись CD/DVD и возможность сохранить за собой коллекцию любимых Windows-приложений. Для новичков или тех, кто ждет стабильной работы прямо «из коробки» – дистрибутив вне конкуренции. А тому, кто постоянно находится в поиске и хочет оставаться на переднем крае новейших технологий, больше подойдет, скорее, что-нибудь из Ubuntu. Или MikeOS. LXF

Свойства навскидку



Легко переключиться

Progression Desktop от Versora переносит ваши персональные данные из Windows в Linux.



Сохраните свои файлы

Функция чтения/записи в NTFS-разделах пригодится, если у вас общие данные для Linux и Windows.

LINUX FORMAT Вердикт

Xandros Desktop 4

Разработчик: Xandros
Сайт: www.xandros.com
Цена: $79 ($39 за Home Edition)

Функциональность	7/10
Цена	8/10
Простота использования	9/10
Документация	9/10

» *Вероятно, лучший дистрибутив для новичков, превосходно документированный, с предсказуемым поведением и хорошими инструментами для миграции.*

Рейтинг 9/10

Cairo 1.2

Грэм Моррисон заблудился на виджетных рынках Каира, он же Cairo – блестящий движок векторного рендеринга, ныне вошедший в Gnome.

LINUX FORMAT на DVD

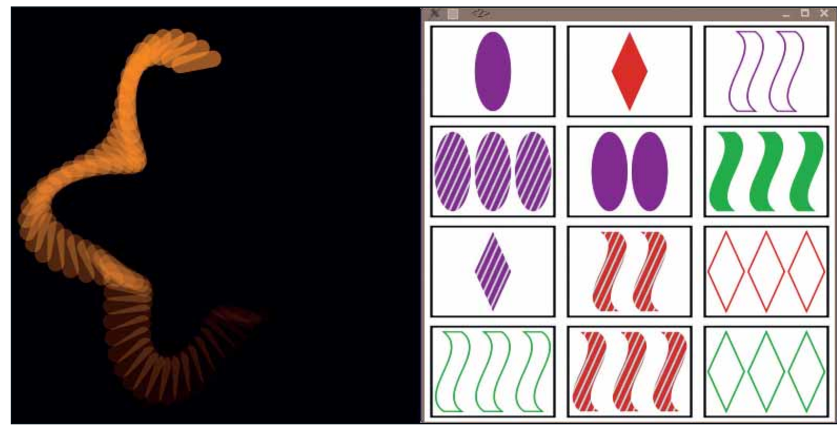
Вкратце...

» Масштабируемая векторная графика для виджетов и окон.

См. также: *Arthur* для *Qt* или *OpenVG*.

С упорством строителей древних пирамид (но, конечно, не из-под палки), команда разработчиков Cairo завершила своё первое крупное обновление. Мы восхищались версией 1.0 еще в LXF74. Cairo – библиотека для создания графики. На первый взгляд, ничего особенного, но Cairo отличается тем, что использует масштабируемую векторную графику вместо растровой.

Векторная графика предпочтительней не только потому, что не связана с конкретным экранным разрешением или соотношением сторон, но и потому, что её обработка может быть ускорена на аппаратном уровне (например, через OpenGL), а при печати она выдаёт наилучшее качество.



» В Сети есть множество программных примеров из Cairo. Скорости хватает для оснащения анимацией (вверху слева), а библиотека графических примитивов почти совершенна.

Cairo 1.2, базируясь на прочном фундаменте прежней версии, развивает скорее стратегическую функциональность, чем внешний лоск. Cairo умеет преобразовывать результаты своей работы в различные форматы, чтобы всевозможные целевые приложения (например, *Adobe Reader* или, что важнее, любая программа *GTK*) могли ими воспользоваться. Наиболее заметная добавка к 1.2 – три новых «выходных интерфейса» (back-end). В первоначальной версии были «интерфейсы» для *Xlib*, с целью применения Cairo в *X*-приложениях; Win32 для платформы Windows; «интерфейс» для создания статических изображений. В версию 1.2 введены «интерфейсы» к PDF, PostScript и SVG (форматам, векторным по своей природе, в отличие от первых трёх), и тут можно ожидать многого. Именно эти три «интерфейса» встроены в принтерный механизм *GTK 2.10* – первой версии, требующей Cairo; итог – немедленное и резкое повышение выходного качества приложений, пользующихся Cairo. *Inkscape* – яркий тому пример, его релиз 0.45 будет использовать исключительно Cairo. Очень важно, что новая версия полностью совместима с предыдущей: обновление существенно упрощается, отпадает необходимость компиляции.

Векторный анализ

«Интерфейсы» к PDF и PostScript (в виде неофициальных дополнений были доступны и для версии 1.0), неизмеримо улучшились – но, что естественно для открытого ПО, наилучшую поддержку имеет формат SVG: ему доступно всё, что может Cairo. Следующим по порядку совместимости идёт PDF, работающий почти так же надёжно. Проблемы возникают только с прозрачностью в PostScript, изображения в котором по умолчанию растровые. Все упомянутые «интерфейсы» теперь полноценны, и возможность выбора между выводом на экран, в SVG и PostScript даёт представление о мощи Cairo.

Разработчики Cairo предприняли немалые усилия по максимальному повышению качества текстового рендеринга, дойдя даже до встраивания шрифтов TryeType в файлы PDF и PostScript для максимальной совместимости. Если шрифты не могут быть встроены, Cairo возвращается к рендерингу текста в кривых. При простом синтаксисе Cairo для рендеринга шрифтов, возможность создания высококачественного текстового вывода как на экран, так и в файл выглядит вполне реальной.

Хотелось бы увидеть прогресс в создании «интерфейса» к *Glitz* для ускорения OpenGL: он всё ещё в экспериментальной стадии, хотя привязки к «языкам» заметно усовершенствованы. В частности, неплохо поддерживается Python, и *Pycario* идет в кильватере обновлений Cairo API; есть также привязка к .NET, включенная в Mono. Caroimm, C++ API для Cairo, почти готов, но его здесь пока нет.

Cairo 1.2 полностью стабильна (благодаря ревизии 1.22), но некоторые операции всё ещё заметно «тормозят» (ещё одна причина стабилизировать «интерфейс» для *Glitz*). Версия 1.2 – солидный шаг вперёд: обновлены и доработаны многие добрые начинания 1.0. Теперь, при достойной поддержке SVG и PDF, пора начинать работу с Cairo. Уже одно это может дать фору Gnome перед KDE 4.0 с его пока неопробованным графическим пакетом, Arthur. LXF

Свойства навскидку



Новые «интерфейсы»

У Cairo теперь есть стабильные «интерфейсы» для SVG, PDF и PostScript.



Обработка текста

Шрифты могут быть встроены в доступные «интерфейсы» или переведены в кривые.



LINUX FORMAT Вердикт

Cairo 1.2

Разработчик: Cairo Development Team
Сайт: http://cairographics.org
Цена: Бесплатно под LGPL

Функциональность	8/10
Производительность	7/10
Простота использования	7/10
Документация	6/10

» *Удивительно зрелая версия, которая только укрепила нашу уверенность в этом превосходном API.*

Рейтинг 7/10

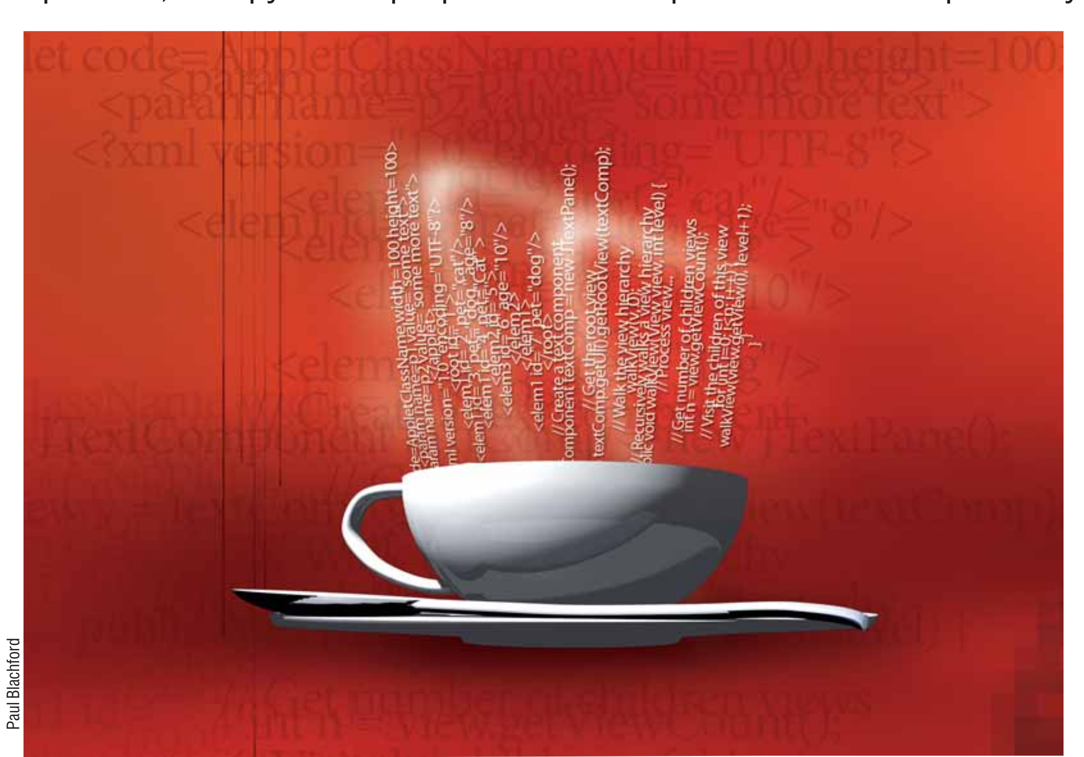
Сравнение



Каждый месяц мы анализируем для вас тонны программ – а вы можете отдыхать!

Среды разработки Java

Ричард Драммонд, подсевший на кофеин, тестирует последние кофеварки – простите, инструменты разработки Java-приложений. Чей аромат лучше?



Paul Blachford

О тесте...

В качестве тестовой системы использовался компьютер Athlon 1,350 МГц с 1 ГБ DDR ОЗУ, под управлением Debian Unstable. Использование такого отсталого процессора выдвигает на передний план показатели производительности IDE, важный критерий в нашем Сравнении.

Поскольку редактор – это основной инструмент в любой IDE, мы тщательно сравнивали среды редактирования, проверяя не просто быстродействие, но и наличие функций, помогающих безошибочно вводить Java-код с наивозможной скоростью. Обращалось внимание на интеграцию со стандартными инструментами, типа *Ant*, *JUnit* и системами контроля версий, на поддержку технологий Java, а именно JSP, JSF и EJB, наличие инструментов визуального проектирования или моделирования, и так далее.

Также учитывалась простота использования, и мы сравнивали, какую документацию, учебные материалы и примеры проектов предоставляла каждая IDE. Наконец, определяющим фактором в рейтинге была цена.

Наш выбор

Пусть это прозвучит еретически, но я хочу сказать «спасибо» Microsoft. Почему? Потому что разработкой и продвижением своей конкурирующей платформы .NET они дали Sun и сообществу Java столь нужный им толчок. Например, с выходом Java 2 Platform Standard Edition (J2SE) 5.0, язык Java наконец-то получил новые возможности, о которых пользователи просили давным-давно, например, поддержку обобщенных конструкций (generics); а предстоящий Java SE 6.0 (кодовое название «Mustang») обещает гораздо большую интеграцию Java-приложений с рабочим столом. В Sun даже посулили открыть исходные коды Java.

Аналогично подогревается и разработка Java-инструментов, подпитываемая главным образом за счет конкуренции между двумя открытыми платформами, NetBeans и Eclipse. Java на световые годы опережает платформу Microsoft в этой области, и Java-разработчикам предлагается гораздо больший выбор, чем их оппонентам на .NET, благодаря большей зрелости и большей открытости платформы Java.

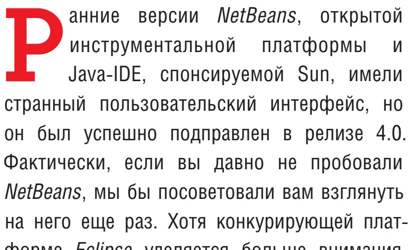
Парад IDE

При столь обширном диапазоне доступных интегрированных сред разработки (Integrated Development Environment, IDE) для Java, по функциям и стоимости – от свободных Java-ориентированных редакторов до дорогих RAD-комплексов уровня предприятия [RAD – Rapid Application Development – средство быстрой разработки приложений, – прим. перев.], кто сможет справедливо сравнить их? Подбирая пакеты для данного Сравнения, мы хотели учесть все многообразие вариантов, так что выбирали в ширину, а не в глубину: мы охватили семь IDE, но, при сложности этих программ, наши статьи о каждой неизбежно коротки.

Мы рассматривали последние стабильные версии каждой из IDE, а не бета- или пререлизные версии. Поскольку каждая IDE включает открытый API для создания расширений, мы ограничились только официальными расширениями и дополнениями, публикуемыми поставщиком – без сторонних или разрабатываемых сообществом расширений.

Здесь пропущена одна заметная IDE: основанная на Eclipse платформа Rational Developer Platform от IBM. Мы хотели включить этот богатый функциями комплекс уровня предприятия, но, к сожалению, не сумели добиться стабильной работы в нашей тестовой системе.

NetBeans IDE



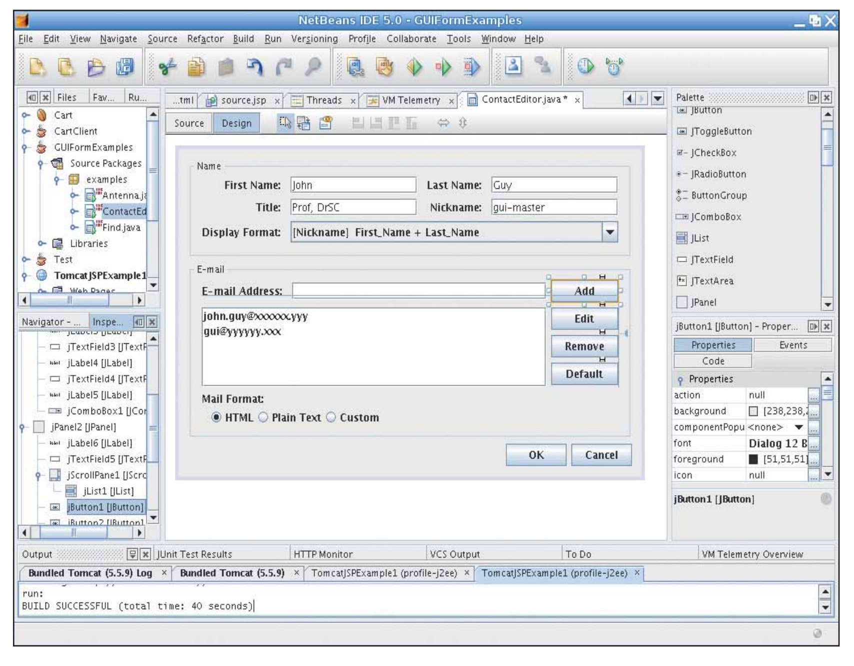
Открытая среда разработки, затмение которой вряд ли произойдет в ближайшем будущем.

Ранние версии *NetBeans*, открытой инструментальной платформы и Java-IDE, спонсируемой Sun, имели странный пользовательский интерфейс, но он был успешно подправлен в релизе 4.0. Фактически, если вы давно не пробовали *NetBeans*, мы бы посоветовали вам взглянуть на него еще раз. Хотя конкурирующей платформе *Eclipse* уделяется больше внимания, *NetBeans* в настоящее время – очень сильный продукт: быстрее, проще в использовании и более совершенный.

Хотя он не идеален. Возьмите редактор NetBeans. Это приятная и быстрая среда для ввода кода Java, HTML, XML, JSP, CSS и SQL, с автодополнением (включая дополнение CamelCase [использование составных слов или фраз, причем слова стыкуются без пробелов, и каждое новое слово начинается с заглавной буквы, – прим. перев.]) и шаблонами. Версия 5.0 наконец-то представляет несколько приличных инструментов рефакторинга, хотя их пока меньше, чем у конкурентов. Редактор распознает ошибки по мере ввода и предлагает варианты для исправления, но эта возможность не столь продвинутая, как инструменты статического анализа IntelliJ. Аналогично, инструменты переформатирования кода в *NetBeans*, на фоне других, не слишком гибки в настройке. Зато *NetBeans* обладает прекрасными инструментами JavaDoc, включая графический интерфейс для быстрого добавления JavaDoc-тэгов в ваш код.

Редактор HTML и JSP не является визуальным. Он предлагает палитру тэгов HTML и JSP, которые вы можете перетаскивать в ваш код для быстрого построения web-страницы. Функция предпросмотра не реализована, так что приходится развертывать страницы и просматривать их в браузере.

Но все однозначно, если говорить о редакторе графического интерфейса. Он всегда радовал, а в версии 5.0 превзошел самого себя. Эта новая версия, под кодовым именем *Matisse*, позволяет вам как попало перетаскивать компоненты мышью на форму и решает, какие менеджеры компоновки лучше всего подойдут для их склейки; имеются динамические направляющие, что гарантирует вам получение элегантных форм с минимальными усилиями. Пользоваться редактором невыразимо просто!



› Дизайнер интерфейса Matisse с его инструментами автоматической раскладки поможет создать красивые формы даже кодерам, лишенным эстетической жилки.

NetBeans интегрируется с встраиваемой базой данных *Apache Derby* и web-контейнером Tomcat. Включена поддержка *JBoss*, *WebLogic* и серверов приложений от Sun, а также различных баз данных, которые буквально штампуются инструментами для создания, просмотра и модификации; также поддерживается создание и работа с web-сервисами. В web-приложениях можно использовать как JSF, так и среду Struts, и, разумеется, *NetBeans* предоставляет инструменты для создания и использования EJB.

Включен интегрированный отладчик, чтобы вы могли отлаживать JSP, сервлеты и EJB, а не просто отдельный код, а монитор HTML помогает при отладке web-приложений. JUnit также поддерживается, что, впрочем, неудивительно.

Совместная работа над кодом

Наряду с этими функциями, *NetBeans* имеет два дополнительных пакета (оба от Sun), которые выделяют его из толпы. Первый – Developer Collaboration. Он добавляет систему мгновенных сообщений (основанную на протоколе Jabber) в IDE NetBeans и инструменты предоставления файлов в общий доступ, что позволяет нескольким разработчикам совместно редактировать исходный файл в одно и то же время. Эти инструменты очень удобны для распределенных групп разработчиков, особенно при пересмотрах кода. Требуется сервер совместной работы, или можно бесплатно подписаться на использование сервиса, предоставляемого на **share.java.net**.

Второй – NetBeans Profiler, включающий нетребовательный к ресурсам Java-профилировщик и различные инструменты для сбора и визуализации полученных данных. Можно выполнить профилирование всего проекта, отдельных классов или просто участка кода, увидеть число созданных потоков, потребление памяти, статистику сборщика мусора... даже подключить профилировщик к уже запущенному процессу (хотя для этого требуется запустить ваш проект в доступной для скачивания модифицированной JVM, виртуальной машине Java).

NetBeans, как и любая программа, не совершенен, но создает ощущение хорошего баланса между удобством, набором функций и ценой (не забывайте, что он бесплатен). Да, редактор мог бы иметь больше излишеств; поддержка систем контроля версий, отличных от CVS, слабовата, и документация не исчерпывающая – но *NetBeans* предоставляет полноценную, всеобъемлющую среду, за которой – целый ряд чудо-инструментов.

Скоро выйдет

NetBeans 5.5 находится на стадии бета-тестирования, но после его выпуска обещан новый пакет дополнений, делающий эту IDE еще более привлекательной для предприятий. В него войдут инструмент моделирования UML (с поддержкой восьми типов диаграмм и одновременного проектирования и анализа [round-trip-engineering]), визуальные инструменты для создания схем XML и проектирования BPEL, а также улучшенная поддержка web-сервисов.



LINUX FORMAT Вердикт

NetBeans IDE

Версия: 5.0

Сайт: www.netbeans.org

Цена: бесплатно под Sun Public License

» Солидное средство для всех типов Java-разработки, с выдающимся редактором GUI и классными инструментами профилирования.

Рейтинг 9/10

Sun Java Studio Creator

Web-приложения – легким движением мыши.

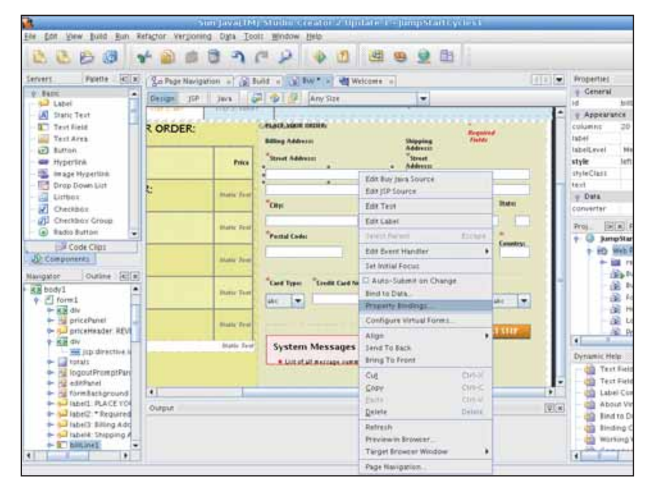
Creator от Sun – это урезанная сборка *NetBeans 4.1* с инструментами для JSF (JavaServer Faces, см. **LXF84**) и разработки портлетов. Она поставляется с Sun *Application Server 8.2* и встраиваемой базой данных *Apache Derby* и формирует полную визуальную среду для разработки web-приложений на базе JSF.

С помощью *Creator* вы создаете JSF-формы, как если бы вы строили «толстые» графические интерфейсы в инструменте RAD – перетаскивая компоненты, манипулируя их свойствами и затем добавляя специфичные для приложения привязки, типа обработчиков событий. Сейчас в качестве слоя представления поддерживается только JSF, но в будущем обещана поддержка Swing. *Creator* дает различные представления форм – представление дизайна, представление JSP и представление Java. Поддерживается связанное редактирование JSF- и JSP-представлений. Предусмотрены инструменты для привязки свойств JSF и для объединения ваших форм с EJB, таблицами баз данных и web-сервисами.

Редактор в *Creator* предлагает те же функции, что и в *NetBeans 4.1*: он быстр, имеет автодополнение кода со встроенным просмотрщиком JavaDoc, и т.д., но его инструменты рефакторинга и форматирования кода скудны.

Есть поддержка баз данных *DB2*, *Oracle*, *SQL Server* и *Sybase*, но Creator интегрируется только с сервером приложений или web-сервером Sun. В отличие от *NetBeans*, интеграция с Ant отсутствует. Инструменты отладки, удобный HTTP-монитор и поддержка среды JUnit упрощают тестирование и отладку ваших приложений.

Creator скорее витрина для JSF, а не практический инструмент, но если вам вдруг захочется разработать что-то с помощью JSF, *Creator* – простейший и быстрейший способ это сделать: он может похвастаться обширной документацией, учебными материалами и примерами кода, а его функция DynamicHelp предоставляет контекстно-зависимые ссылки на справку прямо в основном интерфейсе IDE, так что вы никогда не заблудитесь.



› С такими инструментами визуального проектирования, Creator – единственная IDE в тесте, полностью использующая мощь технологии пользовательского интерфейса JSF.



LINUX FORMAT Вердикт

Sun Java Studio Creator

Версия: Update 1

Сайт: http://developers.sun.com/prodtech/javatools/jscreator

Цена: бесплатно для членов Sun Developer Network (SDN), регистрация в SDN тоже бесплатна.

» *Превосходен для web-разработки на базе JSF.*

Рейтинг 8/10

IntelliJ IDEA

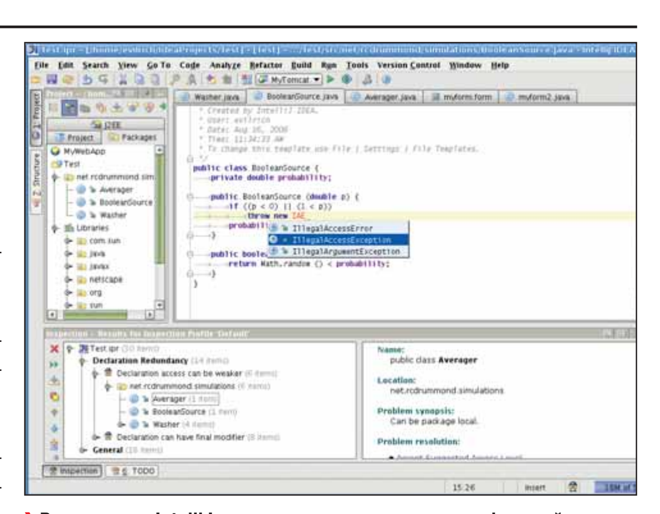
Покорит ли хакеров «интеллектуальная Java-IDE»?

Java-IDE не обязана быть неуклюжей громадиной. *IntelliJ IDEA* – это всего 47 МБ для скачивания, и ее потребности в рабочем режиме тоже скромны. Неудивительно, что в такой размер не вместился ряд инструментов Java EE (Enterprise Edition, или JEE), имеющихся у конкурентов, и связывание с сервером приложений или базой данных.

Где *IntelliJ* выигрывает, так это в гибкой и умной среде, поддерживающей исходный код Java, HTML, XML, JSP, JavaScript и CSS. Наряду с инструментами рефакторинга и генерации кода и всячески настраиваемым инструментом переформатирования кода, имеется и отличный статический анализатор. Например, он подсвечивает неиспользуемые переменные в вашем коде прямо в самом редакторе, в то время как инструмент полного статического анализа информирует вас, где переменные можно сделать окончательными [final], нет ли недокументированных методов, и т.д., и предлагает сделать исправления за вас. Среди недостатков – отсутствие просмотровщика JavaDoc, ассоциированного с инструментом автодополнения кода (фактически, здесь вообще нет внутреннего просмотра JavaDoc), и невозможность предпросмотра страниц HTML и JSP.

Среда разработки графического интерфейса *IntelliJ* поддерживает только Swing, но быстра и проста в использовании. Хотя и не столь блистательная, как *Matisse* из NetBeans, она воспринимается скорее как среда разработки *Qt*-интерфейса: вы размещаете компоненты, где хотите, манипулируете их свойствами, затем группируете их по горизонтальным или вертикальным клеткам или сетке, чтобы построить форму. Интересно, что поведение среды разработки графического интерфейса по умолчанию – генерация из вашего GUI-проекта исполняемого байт-кода, а не редактируемого Java-кода.

Поддерживается интеграция с *WebLogic* и сервером Tomcat, но для других сред JEE вам нужно будет развертывать ваше приложение вручную. Правда, *IntelliJ* сформирует для вас дескрипторы [deployment descriptors] и архив приложения. Серьезное упущение – в *IntelliJ* отсутствуют инструменты для работы с базами данных.



› В редакторе IntelliJ множество вспомогательных функций, повышающих производительность, включая автодополнение CamelCase.



LINUX FORMAT Вердикт

IntelliJ IDEA

Версия: 5.1

Сайт: www.jetbrains.com/idea

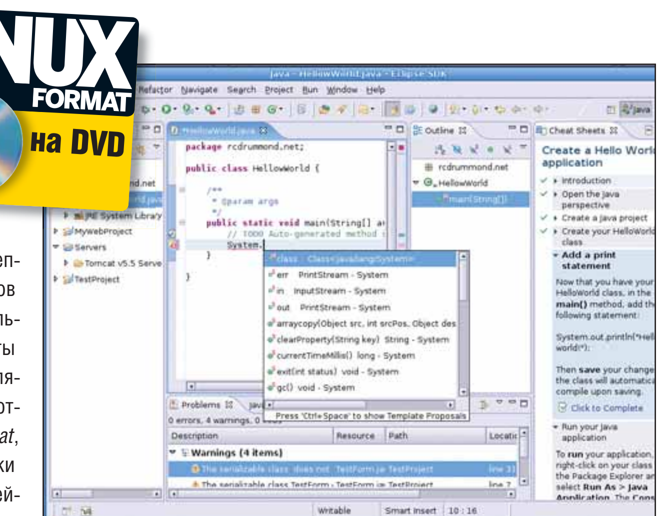
Цена: 499 долларов без НДС

» *Легкая и быстрая IDE для мобильной или настольной разработки, но едва ли пригодная для корпоративных проектов.*

Рейтинг 6/10

Eclipse Callisto

Синхронность между инструментами Eclipse.



› Несмотря на свои недостатки, Eclipse может похвастать рядом изящных функций, вроде этих интерактивных учебников под названием «Cheat Sheets».

Установка *Eclipse*, открытой платформы и Java-IDE, обычно является трудным занятием. Основной проект Eclipse имеет различные подпроекты, например, инструменты для C/C++, графическое редактирование, и так далее. Скачав платформу *Eclipse*, надо разбираться, какие версии инструментов вам нужны для работы с ней. Проект Callisto исправляет это: проще говоря, сделаны скоординированные релизы различных проектов, и ломать голову больше не нужно.

Eclipse отличается от своего конкурента, *NetBeans*, тем, что использует не стандартный набор графических инструментов Java – Swing, а SWT (Standard Widget Toolkit), который, вопреки названию, вовсе не стандартный: это обертка вокруг особого родного инструментария. Например, Linux-версии могут использовать *Motif* или *GTK*. Результат: Java-приложения больше не выглядят чужеземцами на вашем рабочем столе. Второе отличие – по умолчанию Eclipse использует свой собственный инкрементальный Java-компилятор, который жестко интегрирован в IDE, а не стандартный *javac*.

В прежних релизах *Eclipse* был великолепный редактор Java, но не было инструментов решения реальных задач. К счастью, это больше не проблема, поскольку другие проекты *Eclipse* дают свои плоды. Callisto предоставляет инструменты для корпоративной разработки (включая интеграцию с серверами *Tomcat*, *WebSphere*, *WebLogic* и *Oracle*), разработки баз данных, создания графических интерфейсов, тестирования и профилирования.

К сожалению, хотя *Eclipse* теперь вобрал в себя массу Java-технологий, он воспринимается как беспорядочная куча инструментов, а не как интегрированная среда. Его графический интерфейс не лишен внешнего глянца, но навигация зачастую трудна, и *Eclipse* портит впечатление своей вялой производительностью и низкой стабильностью. Тем не менее, эту платформу ожидает хорошее будущее, особенно когда повзрослеют ее подпроекты.

> «Приложения Java больше не выглядят чужеземцами на вашем рабочем столе.»



LINUX FORMAT Вердикт

Eclipse Callisto

Версия: 3.2

Сайт: www.eclipse.org/callisto

Цена: бесплатно под Common Public License

» *Проект выглядит многообещающе, но пока Eclipse лишен зрелости и удобства конкурирующих IDE.*

Рейтинг 6/10

MyEclipse

Eclipse, переделанный для предприятий.

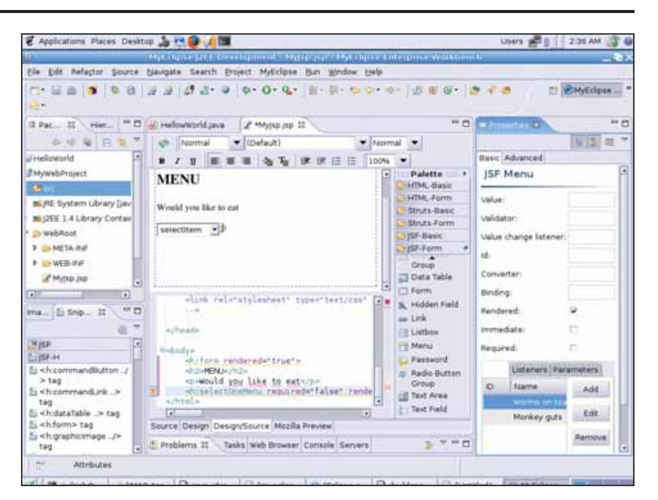
MyEclipse – полностью *MyEclipse Enterprise Workbench* – это набор коммерческих бизнес-инструментов, поддерживаемых как расширения к открытой платформе. Продукт зародился, когда *Eclipse* едва выходил за рамки базовой Java-IDE, но даже сейчас, когда у *Eclipse* есть несколько хороших JEE-инструментов, *MyEclipse* все еще на версту впереди. Он предлагается по подписке, а не в прямой продаже. Полная профессиональная версия стоит чуть больше 50 долларов в год – удивительно хорошее соотношение «цена/качество».

Основанный на той же самой платформе и Java-инструментах, что и *Eclipse Callisto* (см. выше), *MyEclipse*, очевидно, работает похожим образом, так что мы не будем повторяться. Но следует заметить, что *MyEclipse* включает порт восхитительного графического редактора *Matisse* из *NetBeans* в дополнение к тяжеловесному аналогу *Eclipse*. Правда, в настоящее время он поддерживает только компоненты AWT и Swing, но не собственные компоненты SWT *Eclipse*.

MyEclipse отличается от своего открытого родителя своими корпоративными инструментами. В частности, хорошо поддерживается web-разработка, обеспечиваемая в рамках JSP, JSF и Struts, а также Tapestry. Включены визуальный дизайнер процессов, редактор изображений и редактор CSS, но, увы, визуальный редактор страниц в настоящее время доступен только в Windows-версии. Возможна и разработка Ajax, с интегрированным браузером Web 2.0 и «живым» отладчиком JavaScript. Более того, включен продвинутый браузер баз данных и UML-дизайнер, способный производить семь типов диаграмм.

MyEclipse есть за что хвалить, но фундаментальные проблемы IDE *Eclipse* здесь не решены, и остаются те же беды со скоростью и стабильностью. Так же неудобна навигация по интерфейсу; документация местами неоднозначна или вообще отсутствует.

> «Чуть больше, чем $50 в год – удивительно хорошее соотношение цена/качество.»



› Отсутствие визуального редактора страниц означает, что MyEclipse не самое удобное средство web-разработки.



LINUX FORMAT Вердикт

MyEclipse Enterprise Workbench ‘Professional’

Версия: 5.0 GA

Сайт: www.myeclipseide.com

Цена: $52,95 в год без НДС

» *Хорошие J2EE-инструменты, но они не решают фундаментальных проблем платформы.*

Рейтинг 8/10

JBuilder Enterprise

Прежнего лидера рынка теснят конкуренты.

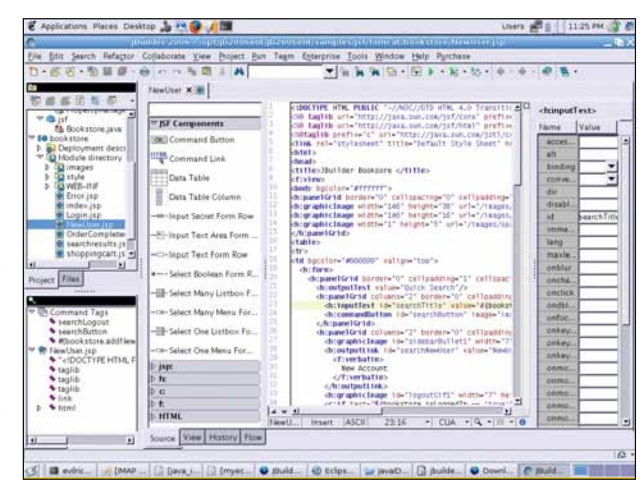
Borland поставляет инструменты разработки с незапамятных времен, и его предложения для Java – функциональные и зрелые продукты. Доступны три версии: бесплатная Foundation edition, Developer edition и Enterprise edition, и в них больше инструментов, чем в вашем ближайшем DIY-центре [Do It Yourself, «сделай сам» – торговые центры, предлагающие широкий ассортимент товаров для дома, – прим. перев.]. Мы займемся, в основном, последней из них.

JBuilder предлагает удобную среду редактирования, со всеми обычными примочками: рефакторингом, переформатированием и инструментами определения и исправления ошибок – хотя его автодополнение кода порой слишком навязчиво, а инструменты *CodeInsight* и *ErrorInsight* не идут ни в какое сравнение с аналогами в *IntelliJ*. Редактор графического интерфейса хотя и удобен, но выглядит седой древностью по сравнению с *Matisse* от *NetBeans*. Из плюсов – Enterprise edition предлагает ряд продвинутых инструментов для аудита кода, отладки и профилирования, и может даже анализировать ваш код на предмет прорех в безопасности.

Основной актив *JBuidler* – ряд его «мастеров» и инструментов, поддерживающих быструю разработку с огромным множеством технологий. Поддерживаются JSP, JSF и Struts для web-разработки. Нужны Cobra или RMI? Они есть. Хотите разрабатывать web-сервисы или EJB? Есть инструмент и для этого. UML? Обижаете! Инструменты совместной работы (a la *NetBeans*)? Естественно.

Неблагодарное занятие – критиковать столь хорошо оснащенный инструментарий, но цена *JBuilder Enterprise* выглядит завышенной – и следует заметить, что эта программа слегка засиделась в первой строчке хит-парада. Ее интерфейс медленный, захламленный и плачет по переделке, а разработка затормозилась, и новые Java-технологии поддерживаются не лучшим образом.

Borland выделила производство своих «традиционных инструментов разработки» в отдельное подразделение и ищет на него покупателя. Если вы подумываете уйти на другую платформу, теперь, возможно, самое время.



› Стойкий и надежный, JBuilder часто проигрывает конкурентам во внешности: пример – его не слишком наглядный JSP-редактор.



JBuilder Enterprise

Версия: 2006

Сайт: www.borland.com/jbuilder

Цена: $3500 (обновление $1900) без НДС

» *Хотя он укомплектован рядом полезных RAD-инструментов, JBuilder ощущает давление со стороны своих открытых конкурентов.*

Рейтинг 8/10

Oracle JDeveloper

Java-IDE от короля баз данных.

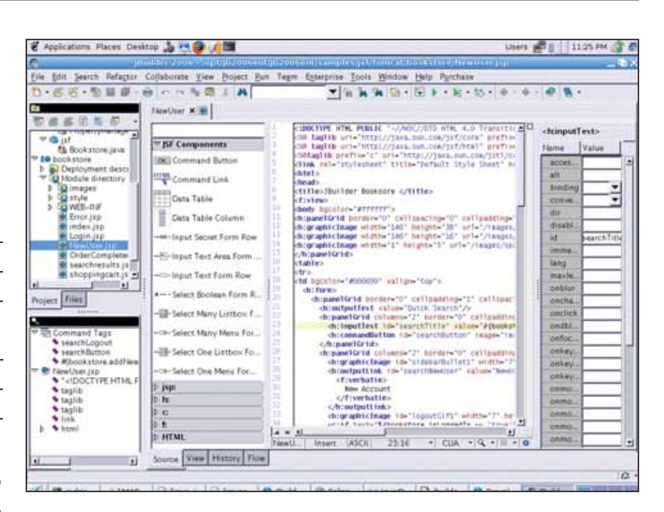
JDeveloper – темная лошадка среди Java-IDE. Он менее известен, чем его соперники, но предлагает сравнимые функции для базового редактирования Java плюс инновационное окружение для разработки J2EE, при превосходных инструментах визуального моделирования и проектирования. Более того, он доступен для бесплатной загрузки.

Корпорация Oracle, несомненно, продвигает *JDeveloper* как приманку, с целью привлечь покупателей к своему серверу приложений, знаменитому серверу баз данных и промежуточному ПО, и не удивительно, что он поддерживает ряд проприетарных технологий, типа Oracle Application Development Framework (ADF) и TopLink (систему постоянного хранения Java-объектов, использующую реляционные базы данных). Тем не менее, *JDeveloper* срабатывается и с другими серверами приложений, включая *JBoss* и *WebLogic*, web-контейнер *Tomcat* и любые базы данных с драйвером JDBC. Приложения ADF можно развернуть на любом J2EE-сервере.

Можете использовать *JDeveloper* как универсальную Java-IDE: здесь в наличии все обычные инструменты редактирования, определения ошибок и рефакторинга, плюс продвинутые инструменты для аудита кода, профилирования, проектирования баз данных и многого другого. Но основное назначение *JDeveloper* – разработка программ на базе ADF. ADF – основанная на MVC среда, упрощающая разработку J2EE-приложений. Она поддерживает для этой модели различные выходные технологии, включая EJB, объекты TopLink и web-сервисы, и можно реализовать просмотр, используя JSP, JSF, Struts или Swing. *JDeveloper* предлагает визуальные и декларативные инструменты для связывания всего воедино.

Oracle снабдил ADF хорошей документацией и предлагает прекрасные учебники и примеры кода, показывающие, как использовать *JDeveloper* с этой технологией для быстрой разработки корпоративных приложений.

> «Основное применение JDeveloper – разработка ПО на базе ADF.»



› В помощь изучению технологий, предлагаемых JDeveloper, Oracle поставляет хорошо документированные примеры кода.



LINUX FORMAT Вердикт

Oracle JDeveloper 10g

Версия: 10.3

Сайт: www.oracle.com/technology/products/jdev/index.html

Цена: бесплатно под закрытой лицензией

» *Скорее RAD-комплекс для Oracle ADF, чем простая Java-IDE, но впечатляющий.*

Рейтинг 7/10

Среды Java-разработки

Вердикт

NetBeans 9/10

Выбор лучшей Java-IDE смахивает на выбор пары обуви. Здесь нет абсолютного чемпиона – все зависит от цели, для которой вы ее обуваете: шлепать по дому, взбираться на горные кручи или танцевать? Так же и проект, над которым вы работаете, определяет, что вам нужно от инструментов разработки. Кроме того, любое решение зависит от личного вкуса. Кто-то может заявлять, что его побитые молью старые валенки – это вершина комфорта, а кто-то – молиться на свои сандалии; разработчик же может считать определенную IDE самой приятной и продуктивной средой для ввода кода независимо от уровня функций, которые она предоставляет.

Выбором *NetBeans* нашей «лучшей IDE» мы объявили, что верим: это лучшая IDE общего назначения по соотношению цена/качество. Нельзя сказать, что она идеальна или лучше всех подходит для какой-то конкретной задачи. *NetBeans* предоставляет комфортную и быструю среду редактирования кода, хотя и не перенасыщенную функциями. Здесь есть первоклассный редактор графического интерфейса, солидные инструменты баз данных, поддержка JEE, достаточная, чтобы сделать жизнь терпимой, профилировщик высочайшего уровня и ряд интересных инструментов для совместной работы. Немаловажно, что *NetBeans* бесплатна, открыта, совместима со стандартами и поддерживается энергичным сообществом.

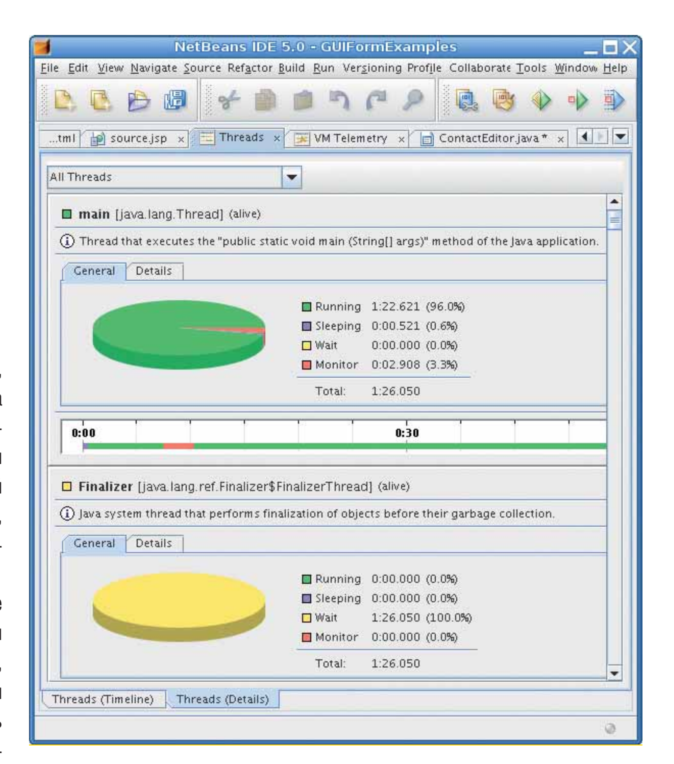
Мы понимаем, что выбрав *NetBeans*, а не ее конкурента *Eclipse*, мы, возможно, бросаем вызов популярному в СМИ мнению. Однако, хотя *Eclipse* имеет задатки классной IDE – и действительно может в один прекрасный день обойти *NetBeans* – она еще таковой не является. Разочаровывает ее нестабильность, низкая производительность и тяжесть интерфейса. Впрочем, вы вправе не согласиться с нами.

Следует помнить, что каждая из IDE в тесте имеет свои преимущества. Инструменты редактирования и статического анализа *IntelliJ* безумно хороши; *Creator* от Sun превращает разработку JSF в плевое дело; *MyEclipse* предлагает ряд первоклассных инструментов по низкой цене; а *JBuilder* просто компетентно все выполняет, если вы готовы платить за это. Поэтому мы настоятельно советуем: если вы приобретаете новую IDE, не принимайте наши слова просто на веру. Все рассмотренные здесь IDE можно бесплатно протестировать; даже коммерческие продукты имеют бесплатные, но ограниченные тестовые версии.

Во введении мы отметили, что рынок Java-инструментов развивается очень быстро. Открытые платформы *NetBeans* и *Eclipse* предлагают даже весьма сложные и зрелые инструменты бесплатно и поднимают планку для своих коммерческих конкурентов. Фактически, уровень конкуренции таков, что даже прежний лидер рынка Borland (разработчик *JBuilder*) решил, что бороться дальше невыгодно. Даже если вы сочтете наш выбор, *NetBeans*, или *Eclipse* пока не пригодным для ваших нужд, оцените эти инструменты еще раз через годик-другой. Вы можете быть удивлены. LXF

Ваше мнение

Вы предпочитаете *Eclipse*, а не *NetBeans*? Какую IDE вы можете назвать лучшим инструментом редактирования или отладки? Или вы думаете, что IDE – это для соляков, и настоящие программеры просто набивают код в *Emacs* или *Vi*? Напишите нам свое мнение на letters@linuxformat.ru.



Дополнения NetBeans по сбору и визуализации профилей производительности работающих приложений великолепны.

> «Eclipse имеет задатки классной IDE, но пока таковой не является.»

Сравнительная таблица функций

IDE	Скорость (5 макс.) Субъективная оценка того, насколько быстро среда работала на нашей тестовой машине	Редактирование (5 макс.) Субъективный рейтинг комфортности и производительности базовой среды	Визуальный дизайнер GUI	Визуальный дизайнер web-страниц	Браузер баз данных	Профилировщик	Инструменты совместной работы	Дизайнер UML	Мобильный SDK
Eclipse Callisto	2/5	4/5	✓	☒	✓	✓	☒	☒	☒
IntelliJ IDEA	4/5	4/5	✓	☒	☒	☒	☒	☒	✓
Sun Java Studio Creator	4/5	3/5	☒	✓	✓	☒	☒	☒	☒
JBuilder	3/5	3/5	✓	☒	✓	✓	✓	2 диаграммы	✓
MyEclipse	2/5	4/5	✓	☒	✓	✓	☒	6 диаграмм	☒
NetBeans IDE	4/5	3/5	✓	☒	✓	✓	✓	☒	✓
Oracle JDeveloper	3/5	3/5	✓	✓	✓	✓	☒	4 диаграммы	☒

Distrowatch



Ежемесячная сводка новостей дистрибутивов Linux.



ЛАДИСЛАВ БОДНАР
основатель, начальник, редактор и сотрудник DistroWatch.com.

Свобода — чего?

Как читатель этой публикации, вы, вероятно, попадали в ситуацию, когда приходилось защитить GNU/Linux, указывая на преимущества свободных программ над коммерческими. Но как убедить вашу организацию хотя бы взглянуть на альтернативную ОС или программу?

Недавняя статья редактора и главы LWN.net, Джонатана Корбетта [Jonathan Corbett], осветила опасности, присущие коммерческому ПО и форматам, описав две реальные истории (см. http://lwn.net/Articles/194701). В обоих примерах поставщики ПО решили увеличить стоимость годичной лицензии и/или поддержки на сотни процентов. В первом случае это было ПО для управления автостоянкой, во втором – программа обслуживания инфраструктуры данных медицинской клиники. Когда ни один из поставщиков ПО не получил увеличенную плату, они заблокировали своих клиентов, что вызвало недовольство автолюбителей, собравшихся покинуть гараж, и лишило врачей доступа к медкартам пациентов.

Открытые взгляды

Отнюдь не каждый поставщик ПО захочет увеличить свои доходы подобным образом, и не всегда есть прямые причины замены коммерческих программных решений аналогами с открытым кодом. Но коли существует альтернатива, то работникам предприятий и организаций, принимающим решения, следует знать о потенциальных ловушках использования закрытых программ и форматов файлов. В конце концов, ставя точку в договоре, они оказываются во власти одной компании и ее прихотей. Почему бы вам не выслать мне свою собственную историю и аргументы защиты?
ladislav.bodnar@futurenet.co.uk

Новая страница

Fedora Core 6 Омолодившись, возвращает поклонников.

Вглядываясь в историю Red Hat Inc. и ее дистрибутивов Linux, легко увидеть синусоидальный характер кривой популярности. Репутация Red Hat рухнула с вершины популярности дистрибутивов Linux на стыке веков, как только фирма попыталась превратить широкую распространенность в звонкую монету, наложив различные ограничения на обновления программ для своей свободной ОС. В 2003 году Red Hat Linux сняли с производства, и ее заменила Fedora Core, что вызвало больше скепсиса в среде ее пользователей. Многие ушли к другим дистрибутивам.



› Fedora Core 6 – один из первых дистрибутивов с Gnome 2.16.

Но после анонса последней тестовой версии Fedora Core 6 появились признаки поворота к лучшему, и Fedora Core вернула большую часть репутации своего знаменитого предшественника. Обзоры двух последних версий (4 и 5) были позитивными, а меры по увеличению сообщества участников проекта Fedora привлекли новых разработчиков и спонсоров. Фирма уделяет намного больше внимания ранее пренебрегаемым аспектам дистрибутива, например, рабочему столу, для чего недавно был создан список рассылки с целью обсуждения идей оформления.

> «Не вызваны ли эти перемены соперничеством Novel и Ubuntu?»

Мы подозреваем, что изменения в сердце Red Hat вызваны острым соперничеством между дистрибутивами от Novel и Ubuntu, более ориентированными на настольные ПК. Хотя ранее фирма утверждала, что Fedora разработан для «технологических энтузиастов» (один из ее руководителей даже докатился до заявления, что для большинства пользователей лучший выбор – Windows), последние ее речи весьма отличаются от этого. «Кто-нибудь непременно заявит, что – просто тестовая версия Red Hat Enterprise Linux. Это неправда, и я не хочу, чтобы впредь так говорили», сказал Уилл Вудс [Will Woods], недавно назначенный руководитель тестирования проекта Fedora. Появление программы Fedora Women и проекта Fedora Usability наглядно подтверждают, что дистрибутив пытается отряхнуть свой старый имидж и привлечь пользователей из различных групп.

Может ли Fedora дошагать до вершины, став лучшим и популярнейшим дистрибутивом Linux? Задача будет трудной. За Ubuntu большая фора и большие деньги, а Novell поставило все, включая будущее компании, на свой продукт Linux. Red Hat не так сильно зависит от Fedora, как Novell от SUSE, поскольку ее дистрибутивы Linux для предприятий сделали Red Hat весьма прибыльным бизнесом. Но если компания из Северной Каролины предоставит своему сообществу Fedora некоторый уровень свободы для внедрения новых идей, без дестабилизации базовых преимуществ дистрибутива, не исключено, что она заставит Ubuntu и OpenSUSE побороться за свои деньги.

Выход Fedora Core 6 назначен на 24 октября, а новая Red Hat Enterprise Linux 5 ожидается следом в декабре 2006.
http://fedoraproject.org

Красивая мечта

Dreamlinux 2.0 Функции и эстетика.

Бразилия, страна Conectiva, Марсело Тозатти [Marcelo Tosatti] и Альфредо Кохима [Alfredo Kojima], вероятно, производит больше дистрибутивов Linux, чем любая другая страна той же площади и того же экономического развития. Но судить о стране только по количеству проектов Linux несправедливо, ибо эстетика рабочих столов делает многие бразильские дистрибутивы произведением искусства.



› **Dreamlinux добился сходства рабочего стола Xfce с Mac OS X.**

Возьмем Dreamlinux. Один из новичков на сцене дистрибутивов Linux, проект недавно вышел в версии 2.0 с ориентацией на настольные компьютеры, под названием Works. Он собран из дистрибутива Morphix на базе Debian, и его можно использовать как LiveCD, но можно также установить на жесткий диск с помощью простого графического инсталлятора. Хотя дистрибутив разработан в Бразилии, команда проекта Dreamlinux изготовила вариант и для англоязычного мира.

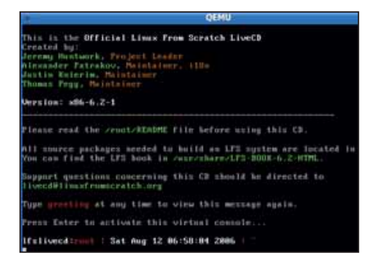
Что такого особенного в Dreamlinux? Кроме использования уникальной реализации *Xfce*, разработчики попытались улучшить впечатление пользователя от дистрибутива с первых шагов, включив компоненты и ПО, не предоставляемое большинством дистрибутивов, например, фото-приложение *Picasa* от Google, графический инструмент *Xara Xtreme* и драйвера Nvidia. Проект также предоставляет версию с *Xgl* для установки на жесткий диск, и это первая реализация 3D-эффектов *Xgl* и *Compiz* для рабочего стола *Xfce*!

Dreamlinux 2.0 Works – превосходный, общедоступный дистрибутив, который, похоже, произведет впечатление на пользователей, подыскивающих хорошо спроектированный рабочий стол с полным набором имеющихся на сегодня бесплатных графических приложений.

Linux с буквой «У»

Linux From Scratch 6.2 Ищущим познания.

Ничто так не повышает самооценку умников-линуксоидов, как успешное «прохождение» книги Linux From Scratch («Linux с нуля»). В разработке с 1999, этот обучающий проект с дистрибутивом типа «Сделай сам» берется преподать пользователям основы Linux и GNU в практичной и развлекательной форме. В августе проект выпустил версию 6.2, включающую ядро 2.6.16, *glibc 2.3.6*, *GCC 4.0.3* и *udev 096*. Книга бесплатна, и предоставляется в нескольких электронных форматах.



› **Linux From Scratch LiveCD – идеальная среда для создания собственного дистрибутива.**

Хотя прочтение 276 страниц чисто технической информации не выглядит веселым занятием, большинство пользователей сумеет справиться с этим за один дождливый уикэнд. Хочется предостеречь: не соблазнитесь простым копированием длинных команд из книги и вставкой их прямо в окно терминала, не глядя на код. Для извлечения из книги максимума действительно необходимо понять каждую команду и каждый ключ, и только потом жать на клавишу Enter.

Исторически Linux From Scratch для использования книги требовал наличия системы Linux с инструментами разработки. В настоящее время достаточно любого полного Live-дистрибутива вроде Knoppix или же родного LiveCD от LFS. Как только LFS будет собран, у вас появится лично ваша система Linux, пусть и несложная. Если вы хотите большего, переходите к Beyond Linux From Scratch (BLFS), который научит вас компилировать огромное число популярных приложений с открытым кодом, включая *X.org*, KDE и Gnome, а также *OpenOffice.org*. Но тут уж надо, чтобы дождь зарядил на неделю или две! LXF

www.linuxfromscratch.org

В разработке... по-прежнему

Большинство дистрибутивов предваряют свой финальный релиз серией рабочих сборок, часто снабжаемых метками ‘alpha’, ‘beta’, ‘test’, ‘pre’, ‘rc’ и прочими. Но какова длительность их циклов разработки? Мы рассмотрели последние выпуски основных дистрибутивов операционных систем Linux и BSD, чтобы понять, сколько времени проходит от выпуска первой рабочей сборки до финального релиза. Интересно отметить, что чем дистрибутив «хитроумнее», тем менее вероятно, что он предоставляет полные рабочие сборки – Debian, Gentoo и Slackware все попадают в эту категорию, и есть только один способ их протестировать: запустить их разрабатываемые ветки, поскольку они медленно сходятся к стабильному релизу.



› **В Damn Small Linux всего 50 МБ, отсюда и промежуток в три недели между сборками.**

Версия дистрибутива	Первая сборка	Стабильный релиз	Промежуток в днях
SUSE Linux 10.1	29/9/05	11/5/06	224
Ubuntu 6.06	19/11/05	1/6/06	194
Red Hat Enterprise Linux 4	28/9/04	15/2/05	140
Fedora Core 5	23/11/05	20/3/06	117
SimplyMepis 6.0	10/4/06	22/7/06	103
FreeBSD 6.1	9/2/06	9/5/06	89
Mandriva Linux 2006	26/7/05	6/10/05	72
Freespire 1.0	17/7/06	8/8/06	22
Damn Small Linux 3.0	30/5/06	20/6/06	21

Хит-парад дистрибутивов

10 самых посещаемых страниц на DistroWatch.com за четыре недели с 01/08/06 по 29/08/06 (среднее число визитов в день)

	Дистрибутив	Количество визитов	
1	Ubuntu	2,403	‹›
2	SUSE	1,790	‹›
3	PCLinuxOS	1,299	↑
4	Fedora	1,182	↓
5	SimplyMepis	1,002	↓
6	Mandriva	869	‹›
7	Slackware	800	↑
8	Damn Small Linux	756	↓
9	Debian GNU/Linux	740	↓
10	Gentoo	574	↑

» DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве посещений сайтов, посвященных конкретным дистрибутивам. Хотя эти цифры и не отражают реальное количество инсталляций, они являются индикатором популярности дистрибутива на данный момент времени.

Linux-предприятие от Novell



Спустя три года после того, как Novell пришла в Linux, купив SUSE, **Майк Сондерс** и **Грэм Моррисон** расследуют, работает ли ее новая стратегия.

Для проникновения Linux в большой бизнес нужна поддержка большого бизнеса. Мир хакеров, живущих в плохоньких комнатушках, и любительских проектов SourceForge, возможно, превосходен для развития новых идей и талантов разработчиков, но если вы хотите развернуть Linux на своем предприятии, вам нужно большее: нужна поддержка 24/7, нужны люди на телефоне для случаев, когда что-то идет не так – и более того, нужны гарантии, что ваши Linux-инсталляции будут поддерживаться несколько лет. Конечно, открытость кода означает, что при возникновении проблемы вы можете исправить ее сами, но не каждая фирма может позволить себе нанимать хакеров на полную ставку.

Когда Linux добрался до предприятий, мы увидели, что фирмы, подобные Red Hat и IBM, предлагают широкомасштабную поддержку и услуги по разработке для этой ОС. Это не только аргумент для ИТ-менеджеров – это также помогает утверждать Linux в «реальном мире», доказывая, что это не какая-то рискованная технология-однодневка, но что на нее можно положиться. Linux уже здесь, готов к использованию, и становится все сильнее.

> «За Linux стоят крупные фирмы, значит, это не преходящая технология.»

Мы склонны считать IBM и Red Hat основным дуэтом поддержки Linux, но Novell имеет свои виды на корпоративный рынок Linux. Закупив Ximian и SUSE, Novell приобрела очень богатый опыт в Linux, как для настольных систем, так и для серверов – и с ее историей NetWare и корпоративными сервисами, довольно убедительно претендует на звание главного игрока в мире Linux, причем на долгие годы. Многие продолжают считать Novell новичком в Linux (название Novell якобы было предложено женой основателя, которая думала, что это означает «новый» по-французски), но, как мы знаем, в компьютерном мире все меняется со скоростью света.

Последние релизы SLED 10 и SLES 10, соответственно настольного и серверного продуктов Novell, породили серьезные дискуссии в Linux-сообществе – но что они действительно предлагают? Как выглядит Novell по сравнению с Red Hat, насколько успешна в этой области? И какая часть постоянно растущего стека ПО Novell имеет смысл в использующем Linux корпоративном окружении? Читайте дальше, чтобы узнать все подробности...

На этих страницах

Как NetWare встретилась с SUSE

Как пользователи Linux, мы теперь воспринимаем присутствие Novell в мире Open Source и на Linux-сцене как само собой разумеющееся. К примеру, это был один из самых заметных экспонентов на последней выставке LinuxWorld в Сан-Франциско. Но Novell, ставшая Linux-предприятием три года назад, приобретя в 2003 г. SUSE Linux за 210 млн. долларов, рассматривает Linux лишь как новейшую историю. На самом деле фирма вступила в жизнь как скромный изготовитель аппаратуры в Прово, штат Юта, в 1979 г.

Novell стала известна благодаря NetWare, сетевой операционной системе, которая доминировала на рынке сетей в начале 90-х (и, бесспорно, продолжает оставаться самым известным продуктом Novell). В течение десятилетия доля рынка NetWare разъедалась операционной системой Microsoft Windows NT, привлекающей все больше системных администраторов благодаря своей интеграции с другими продуктами Microsoft. Сетевые администраторы держали NetWare на хорошем счету, но при убывании поддержки от сторонних организаций, приток потенциальных потребителей мельчал.

Вышедшая в 2001 г. NetWare 6 должна была стать последним поколением самостоятельного продукта NetWare. Но многие из оставшихся пользователей были важными клиентами, и в 2003 была анонсирована замена, названная Open Enterprise Server (OES). Она готовилась к выпуску еще два года, но при этом включала уже два ядра: SUSE Linux Enterprise Server (SLES) 9 как Linux-компонент, и NetWare 6.5. Это ознаменовало новое направление Novell. В августе 2003 года вице-председатель Novell Крис Стоун [Chris Stone] отозвался о Linux как о «гигантской возможности» – шансе построить полный стек программного обеспечения, сравнимого с Microsoft-овским.

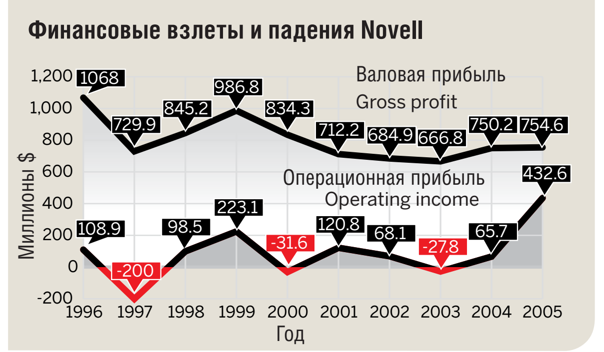
«Как вы могли заметить, доля приложений и сервисов для Linux в корпоративном секторе весьма незначительна,»–сказал Стоун в интервью CNET News.com. «За 15 с лишним лет Novell построила очень устойчивые приложения и услуги корпоративного класса, и ничего из этого не существует сегодня в полностью коммерческом виде для Linux. Потребители говорят нам, что они хотят перейти на другую платформу – на эту платформу». В том же месяце Novell купила Ximian, разработчика Gnome и Mono, а в ноябре 2003 г. анонсировала планы приобрести нюрнбергскую фирму SUSE Linux.

Alex Williamson

SUSE возник как перевод Slackware на немецкий язык, и был продуктом консалтинговой Unix-группы с тем же именем. Дистрибутив стал невероятно популярен, особенно в Европе, и его репутация как стабильной и функциональной ОС сохранилась при переходе в Novell. В Open Enterprise Server Novell смогла объединить свою платформу NetWare с Linux-платформой, переводя пользователей NetWare на путь обновления и в то же самое время давая понять, что ее будущее принадлежит Linux.

Тень Microsoft

Novell рассматривает Microsoft как своего основного конкурента, и SLES – попытка вернуть часть клиентской базы, потерянной со времен расцвета NetWare. В документе, опубликованном в июле 2005 г., Грег Манкузи-Унгаро [Greg Mancusi-Ungaro], директор Novell по маркетингу, Linux и Open Source (см. наше интервью в LXF80), описал, что он счел основными причинами выбора SLES. И основная цель – давний соперник Novell, Windows NT. Грег хотел, чтобы потребители заменили весь сетевой стек NT – *Microsoft Exchange*,



хостинг приложений, управление файлами и печатью и другие сетевые сервисы – на SLES, который он рекламировал как стабильную, дешевую альтернативу NT. Вот так Novell видит Linux, и Linux дал Novell очень нужный толчок, снова выпихнувший ее на арену, где она была десять лет назад.

»

Этапы развития Novell

1979	1983	1989	1993	1994	1996	1999	2003	2004	2005
Основана Novell Data Systems для производства систем на базе CP/M.	Переименована в Novell Inc., запущена сетевая ОС NetWare.	Выпуск 32-битной версии NetWare.	Novell приобретает Unix System Laboratories за $ 332 млн.	Приобретение WordPerfect Corporation, с ранним GroupWise.	Компания делает ставку на «интернетизацию» своей продукции.	Выпущен сервер каталогов eDirectory.	Novell покупает Ximian, спонсирующую разработчиков Mono и Gnome.	Дальнейшее приобретение SUSE Linux и ее дистрибутива.	Запущен проект OpenSUSE.



Деловые опции

Осмысление стека ПО Novell, использующего новые технологии SLES 10 и SLED 10.

Alex Williamson

На первый взгляд, набор программ, предлагаемых Novell, как и в случае других Linux-компаний, нацеленных на предприятия, сбивает с толку – есть различные платформы, двусмысленно определенные как промежуточные (middleware) решения и поддерживающие сервисы, которые объединяют части в единое целое. Novell подошла к корпоративному Linux с двумя основными продуктами ОС: SLED (SUSE Linux Enterprise Desktop), клиентской ОС, используемой на технических рабочих станциях и офисных настольных компьютерах; и SLES (SUSE Linux Enterprise Server), спроектированным для различных задач: от сервера уровня отдела до хостинг-серверов с высоким уровнем устойчивости к стрессовым нагрузкам.

К этим ОС можно добавить другие функции, как мы скоро увидим – а пока займемся предложениями основной ОС.

Рабочий стол для работы

SLED 10, по цене за обновления 50 долларов в год на одного пользователя (или 125 долларов на три года), во многом основан на Novell Linux Desktop, предыдущем клиентском Linux-продукте Novell, ориентированном на предприятия. Он вобрал в себя новый код и функции проекта OpenSUSE и собственные наработки Novell, например, Xgl, и доступен для свободной загрузки (вам нужно будет получить регистрационный ключ для обновлений в течение 60 дней – сейчас это бесплатно). Если вы фанат SUSE, вы встретите в SLED много старых друзей; он все еще использует всеобъемлющий *Yast* как основной инструмент настройки, и надежность всего дистрибутива очень высокая, как пользователи SUSE и ожидали.

Однако есть немало отличий: акцент сделан на корпоративное применение, и рабочим столом по умолчанию является Gnome, если не выбрать другой во время установки. «И для нас, и для пользователей проще иметь один рабочий стол по умолчанию», поясняет Мартин Лазарх [Martin Lasarch] из команды SUSE Core Systems. «С тех пор как Novell купила Ximian, а потом SUSE, у нас больше разработчиков Gnome, чем KDE. Мы вкладываем в Gnome много труда, и нам проще сопровождать его как устанавливаемый по умолчанию. Конечно, поддержка KDE продолжается, и команды работают вместе, улучшая их оба».

Сам Gnome усилен дополнительным «Главным Меню», заменой традиционного меню приложений: здесь есть интегрированный поиск *Beagle* и быстрый доступ к инструментам настройки. Для поддерживаемых видеокарт Xgl вызывается одним щелчком – можно поспорить, нужны ли дистрибутиву Linux для корпораций супер-яркие функции, типа рабочих столов в виде трехмерного куба, но это, конечно, добавляет «вау»-фактора при демонстрации SLED пользователям Windows. А достаточно ли надежен Xgl для корпоративных настольных систем? «Мы считаем, он готов к работе, и мы, естественно, тестировали его у себя перед выпуском», уверяет Лазарх. «Мы использовали его для наших презентаций SUSE Linux 10.1 и SLED 10 на CeBIT и LinuxTag в Германии, и на протяжении всего мероприятия проблем не было, а это было задолго до релиза».

Для основных задач системного администрирования используется *Yast*, хотя в SLED он редко запускается как отдельная программа: вместо этого есть ссылки на его подинструменты из Центра Управления (также обслуживающего запуск всех диалогов настроек



› **Рабочий стол SLED 10 по умолчанию использует Gnome (но поддерживается и KDE) и может похвастать новыми умными меню и диалогом настройки.**

Тренировки на марше

Со SLES и SLED 10, Novell представила необычную тренировочную систему, чтобы ускорить освоение этих ОС администраторами. Вместо выдачи обширного списка учебников, курс First Class от Novell предлагает маленький портативный USB-диск, содержащий готовые дисковые образы SLED и SLES, наряду с *WMware Player*. Эти ОС можно попробовать прямо в Windows или Linux, без предварительной инсталляции.

Спроектированная как курс для самообразования, система First Class в настоящее время находится в бета-стадии и доступна на сайте Novell (www.novell.com) за 199 долларов. Администраторы и инструкторы, желающие продвинуться дальше, могут выбрать другие учебные материалы, например, о сетевых сервисах и безопасности, цены на которые начинаются с 396 долларов.

❯ SLES 10 и SLED 10 разделяют один и тот же инсталлятор – не самый быстрый в мире, но удобный для работы.

рабочего стола Gnome). Все это построено на ядре 2.6.16.21 и *glibc 2.4*, с *X.org 6.9.0*, *Gnome 2.12.2*, *OpenOffice 2.0.2*, *Firefox 1.5.0.4* и *Evolution 2.6.0* в качестве основных компонентов. Также доступен KDE 3.5.1, но SLED не устанавливает инструментарий GNU по умолчанию, сохраняя размер нормальной инсталляции менее 2.8 ГБ. В Gnome также включены различные приложения Mono, например, заметки Tomboy, аудиоплейер *Banshee* и менеджер фотографий *F-Spot*.

По большей части, в SLED не так много нового, чего не найдешь в OpenSUSE и связанных с ним репозитариях – различия заключаются в демо-шлифовке внешнего вида и долгосрочной поддержке. Релизы OpenSUSE поддерживаются два года, а Novell будет предлагать обновления SLED до июля 2011 г. Итак, OpenSUSE не исключается развертыванием SLED/SLES на предприятиях, но для машин, нуждающихся в надлежащей поддержке и долгосрочных обновлениях, корпоративные продукты – лучший выбор. Novell сейчас позиционирует SLED как решение для технических рабочих станций и офисных сотрудников, а «Novell Linux POS» - для касс, стоек регистрации аэропортов и т.д.

Какой SUSE что делает?

» OpenSUSE – разрабатываемый сообществом дистрибутив, с регулярными (шестимесячными) релизами. Предназначен для пользователей домашних систем, увлеченных людей и разработчиков, желающих самостоятельно исправлять проблемы.

» SUSE Linux – версия OpenSUSE. Бесплатна для скачивания, коробочная версия доступна за 38,95 долларов с 60-дневной поддержкой установки от Novell. Спроектирована для домашних пользователей, рабочих станций разработчиков и малых серверов.

» SUSE Linux Enterprise Server/Desktop – дистрибутивы уровня предприятия с акцентом на безопасность и стабильность. Предлагают долгосрочную поддержку



Novell Inc

Серверные спецификации

SLES 10, серверный дистрибутив, доступен в пределах цен от 349 долларов за один год обновлений до 3748 долларов за три года приоритетной поддержки, обновлений и обучения (с различными промежуточными вариантами). Как и в случае с SLED, можно бесплатно скачать SLES 10 с сайта Novell, но чтобы получать надлежащую поддержку, необходимо купить одну из описанных выше опций. SLES основан на той же кодовой базе, что и SLED – соответственно, имеет ядро 2.6.16.21 со множеством пакетов (SMP, Xen и т.д.), наряду с *glibc 2.4*.

Основные серверные компоненты включают *MySQL 5.0.18*, *PostgreSQL 8.1.4, Apache 2.2.0*, *Sendmail 8.13.6*, *Postfix 2.2.9* и *Bind 9.3.2*. По части скриптов вы найдете Perl 5.8.8, PHP 5.1.2 и Python 2.4.2, и все это собрано с помощью *GCC 4.1.0*. Фирмы обычно используют SLES как базу для ряда других серверных продуктов Novell (*см. ниже врезку «Стек SLES»*). »

Стек SLES

Серверный «стек» Linux от Novell состоит из нескольких компонентов, работающих на SUSE Linux Enterprise Server, с прослойкой *AppArmor* для дополнительной безопасности. Выбор компонентов гибок, но здесь мы сосредоточимся на трех основных серверных продуктах, предлагаемых Novell.

GroupWise

Это программный пакет от Novell для групповой работы – конкурент *Microsoft Exchange* и *Outlook*. Он совмещает электронную почту, общий календарь, функции мгновенных сообщений и управления документами, с серверной частью, работающей на SLES, NetWare или Windows, и клиентами, доступными для Linux, Windows и Mac OS X. Пользователи настольных систем, не запускающие клиент *GroupWise*, по-прежнему имеют доступ к серверу через расширения POP3/IMAP или web-интерфейс. Разработанный корпорацией WordPerfect, он был объединен с eDirectory, когда Novell приобрела WordPerfect в 1994 г.

ZENworks

ZENworks спроектирован для помощи предприятиям в управлении всеми своими машинами – настольными системами, серверами и ноутбуками под управлением Windows и Linux. ИТ-администраторам здесь предлагается удаленное управление машинами для установки ПО, устранения проблем и наложения исправлений. Дополнительно, он упрощает миграцию Windows-машин (т.е. если пользователь переходит с одного Windows-компьютера к другому, его настройки и документы должны плавно переместиться). По части Linux, приобретение фирмы Ximian дало Novell инструмент управления пакетами *Red Carpet*, на базе RPM.

eDirectory

Это сервер каталогов для организации и управления сетевыми ресурсами (общими папками, аутентификацией пользователей и групп, принтерами и т.д.), ранее известный как Novell Directory Services: Novell-эквивалент *Active Directory* от Microsoft и сервера *Fedora Directory* от Red Hat. Для очень больших корпоративных инсталляций eDirectory можно настроить, чтобы «реплицировать» часть каталога на другие машины и снизить загрузку отдельного сервера.

Novell AppArmor

Соперник SELinux, этот комплекс безопасности был первоначально разработан Immunix, ставшей частью Novell в 2005 г. AppArmor позволяет администратору назначать профили безопасности с программами, определяя, что они могут, а что – нет (например, получать доступ к определенным файлам или открывать сетевые порты). Он основан на интерфейсе ядра LSM (Linux Security Modules); за дополнительной информацией обращайтесь к **LXF83**.

SUSE Linux Enterprise Server

Примеряем Red Hat

Если вы продаете Linux большому бизнесу, рано или поздно вы столкнетесь с «красной фетровой шляпой».

Не секрет, что главный соперник Novell в битве за корпоративные доллары – это Red Had, фирма из Северной Каролины, доминирующая в Linux-бизнесе все 13 лет после своего основания. Пока что на бумаге это может выглядеть так, как будто нет повода для спора в вопросе доминирования на рынке. В Novell работает 5700 человек, в то время как в Red Hat менее 1150 сотрудников. Годовой доход Novell в 2005 году достиг 1197 млн долларов, в то время как годовой доход Red Hat составлял шестую часть от этого – чуть меньше 200 млн долларов.

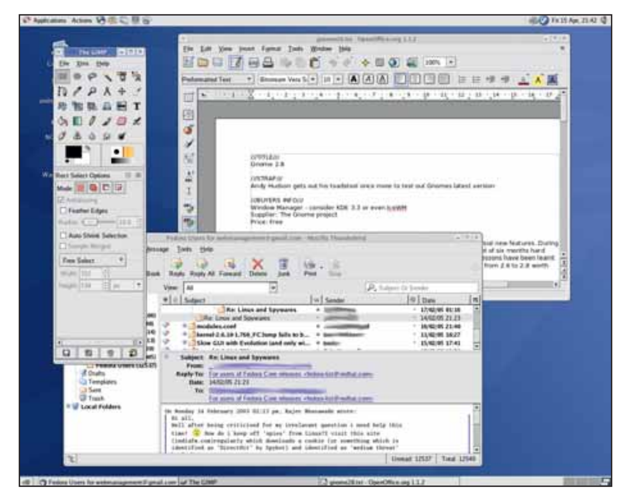
Однако Novell все еще делает большую часть этих денег на NetWare. Когда речь заходит о Linux, Red Hat – несомненный король, с почти 35% рынка web-серверов, по данным онлайн-статистики Netcraft (доля серверов на базе дистрибутивов SUSE – 12%). Но чем действительно разнятся эти две фирмы на корпоративном рынке Linux, так это позициями, занимаемыми ими по отношению к революции ПО с открытым кодом.

Red Hat выглядит как типичный open-source-бизнес. Он улучшает быструю модель разработки, характерную для open-source-сообщества, передавая преимущества, которые дает эта модель (частые релизы и новейшие функции) своим потребителям. Когда бизнес думает о Linux, он обычно подразумевает Red Hat. Novell возгорелась из тлеющих угольков своей проприетарной бизнес-модели, и хотя она сопровождает некоторые наиболее популярные проекты с открытым кодом, она все еще опасается открывать исходный код проектов и бизнеса, за которые выложила хорошие деньги. В прошлом она называла себя компанией со «смешанным кодом»; возможно, это делает ее Lee&Perrins в мире Linux.

> «Когда бизнес думает о Linux, он обычно подразумевает Red Hat.»

Хорошей иллюстрацией различий в философии бизнеса между Red Hat и Novell служит их подход к новым приобретениям и их применению. В конце 2004 года Red Hat купила хромающий Netscape Server и Netscape Certification у AOL, примерно за 22 млн долларов. Шесть месяцев спустя, как и было обещано, это ПО стало составной частью корпоративных предложений Red Hat, возродившись в виде Red Hat *Directory Server* – платной альтернативы открытым решениям вроде OpenLDAP. Но более важно, что Red Hat выпустила исходный код под GPL и основала проект *Fedora Directory Server* как опирающуюся на сообщество платформу разработки, связанную с дистрибутивом Fedora. Пользователи теперь знакомы с моделью *Red Hat* – плати за корпоративную версию или скачивай бесплатно версию, сопровождаемую сообществом.

И это не единственный пример, когда Red Hat покупает технологию, а затем выпускает ее на свободу. В начале 2004 года Red Hat купила Sistina Software и выпустила ее основной продукт, Global File System (GFS), под GPL, в то время как корпоративные потребители платят 2200 долларов за версию с гарантией поддержки. Исполнительный директор Red Hat Мэтью Зулик [Mattew Szulik]), сказал, что подобные приобретения позволяют Red Hat «глубже проникать на корпоративный и правительственный рынки». Крупный капитал хочет убедиться в жизнеспособности ПО, прежде чем платить за экспертную поддержку. А кто более подходит для предостав-



› В отличие от SLES, открытая природа Red Hat привела к появлению множества клонов его корпоративного дистрибутива, включая CentOS.

ления такой поддержки, как не фирма, владеющая технологией и большинством разработчиков?

Открытый или закрытый

Novell, напротив, имеет тенденцию держать приобретенные технологии ближе к кубышке. Целый ряд их старейших технологий все еще проприетарен – из них наиболее известен сервер каталогов *eDirectory*. Это прямой конкурент *Directory Server* от Red Hat, но не открытый. Novell открыла некоторую его часть, например, интерфейсы разработчика, позволяющие сторонним приложениям проходить аутентификацию в eDirectory, но это совершенно иной подход, чем у Red Hat.

Хотя приобретение Novell SUSE и Ximian принесло свои плоды сообществу Open Source, ни тот, ни другой не требовалось выпускать из оков проприетарной лицензии. Но Novell, похоже, решила изменить тактику. Фирма унаследовала приложение безопасности *AppArmor*, когда приобрела Immunix в конце прошлого года, и никто не мог обвинить ее в сдерживании, когда она выпустила *AppArmor* как открытое ПО в начале 2006 года. Novell также начинает высказываться против проприетарных модулей (например, драйверов некоторых графических карт), которые связаны с их Linux-дистрибутивами. Пользователи теперь должны будут использовать сценарий автоматического скачивания, чтобы загрузить модули с сайта производителя, а не с инсталляционного носителя. Компания, видимо, прислушалась к влиятельным разработчикам ядра, вроде Грега Кроа-Хартмана [Greg Kroah-Hartman], занятого в SUSE-лаборатории Novell: он всегда заявлял, что запуск проприетарных модулей с вашим ядром уже нарушает положения GPL (*см. наше интервью в* LXF81).

Novell предстоит еще долгий путь в сообществе Open Source, прежде чем люди убедятся, что она так же предана движению, как Red Hat. В настоящее время может скорее показаться, что Red Hat лидирует, а Novell идет следом. После запуска проекта Fedora сдвиг от коробочных версий SUSE к разрабатываемой сообществом OpenSUSE воспринимался некоторыми как ответный ход, а не инновация.

Но большинство предприятий не волнует отношение фирмы к Open Source, и в любом случае они не против платы за поддерживаемые версии. Сможет ли Novell бросить вызов Red Hat на корпоративном рынке, зависит от слаженности работы элементов ее стека ПО и от качества ее технологии. »

SLES против RHEL

Цифры

Основная	RHEL	349 долларов (максимум два ядра [CPU])
	SLES	349 долларов (неограниченное число ядер)
24-7	RHEL	2499 долларов (только AS Premium Edition)
	SLES	1499 долларов

Аппаратные платформы

И Red Hat Enterprise Linux, и SUSE Linux Enterprise Server поддерживают x86, AMD64/EM64T, Itanium II, Power и zSeries.

Поддержка

Если с сервером что-то пойдет не так, эффект может быть катастрофическим. Вот почему поддержка столь важна. Чем быстрее инженер Novell сможет отреагировать на проблему, тем меньше денег потеряет ваш бизнес. 24-часовая поддержка семь дней в неделю стоит дорого, но для большого бизнеса это разумно вложенные деньги. Если ваш сервер не столь критичен, вы можете обойтись базовой поддержкой. Это означает, что Novell будет помогать вам 90 дней, на время установки сервера, и затем будет предоставлять исправления и обновления операционной системы целый год.

На что похож SLES? Социологическое исследование

Мы расспросили Алексея Рудева [Alexei Roudev], ведущего системного и сетевого инженера консалтинговой фирмы Exigen USA, о его опыте работы со SLES на протяжении трех лет.

Linux Format: Почему вы выбрали Novell, а не Red Hat?

Алексей Рудев: Мы выбрали SLES вместо RHEL для новых проектов пару лет назад, отчасти по личному предпочтению, но на него воздействовали многие факторы. В то время SUSE 8 и 9.3 предлагали последовательную и завершенную систему, со всеми необходимые частями, такими, как *Yast*, KDE и *You* – система обновлений и установки по сети. RHEL выглядел полу-Ferrari, полу-паровозом. Прекрасный пример – эти жуткие «друиды» в RHEL. Также повсеместно различались установка и инструменты администрирования.

Затем мы начали искать серверную платформу. Нам нужна была поддержка *Oracle*, *Real Applications Cluster* (RAC), iSCSI (интернет-хранилище) и Fibre Channel. SLES 8 был единственной системой, имевшей интегрированный *Oracle RAC* и интегрированную документацию Oracle, и все драйверы (iSCSI, FireWire для наших тестов, даже некоторые из Fibre Channel), были также интегрированы. С RHEL мне понадобилось бы многое скачивать вручную, устанавливать и компилировать, прежде чем это заработает. В обновлении SLES 8 даже есть встроенная OCFSv1, позволяющая запускать *Oracle RAC* без всяких ручных изменений. Мы выбрали Novell и подписали партнерское соглашение, разрешающее нам делать свои собственные изменения.

LXF: Многое ли поменялось с тех пор как вы впервые начали использовать SLES?

AP: Novell прекратила поддержку SUSE 8, что стало для нас началом года головной боли, поскольку SUSE 8 была (и остается) единственной системой, поддерживающей все версии Oracle при полной поддержке производителя, Oracle Corporation. SLES 9 стал первым коммерческим Linux на ядре 2.6. Он сохранил все преимущества старого SUSE в отношении Oracle, и сертификация других приложений появилась быстро. Вначале релиз SLES 9 был не очень удобен и создавал нам серьезные проблемы с производительностью, но выпуск Service Pack 1 решил большинство проблем. Мы в конечном итоге перешли на SLES 9 SP1 и запустили на нем несколько подготовительных и разрабатываемых проектов (включая кластер *Oracle RAC* в тестовой лаборатории). Мы настояли на использовании SLES 9 для новых проектов, и это со временем окупилось. Теперь у нас есть поддержка производителей (например, Open Manager от Dell) для SLES 9 SP3, у нас есть очень стабильный iSCSI с Oracle на нем, стабильный HA-кластер с OCFSv2 поверх него. Очень важно, что это все получено «из коробки», хотя нередко и в сильно устаревшем виде. Например, OCFSv2 имеет собственную ветвь в SUSE версии 1.2.1, в то время как оригинальный открытый проект устранил множество ошибок и находится на версии 1.2.4.

LXF: Вы планируете переходить на версию 10?

AP: Я тестировал SLES 10, начиная с бета-версий. Идея хорошая, но реализация все еще в развитии. Система обновлений нарушена (*You*), и ее замена пока еще не работает должным образом, и использует несовместимую систему, которая делает нашу инфраструктуру бесполезной. Также есть ряд серьезных проблем с инсталляцией (SLED 10 не работает с некоторыми CD-приводами и считывает некорректные данные), рабочий стол Gnome, используемый теперь по умолчанию, имеет бедные настройки, которые сбивают людей с толку, и *Yast2* имеет обычный набор небольших проблем. Мы решили подождать Service Pack 1.



› **«Игрок команды Novell»** (справа). **Люди вроде Алексея успешно используют SLES в серверных, и хотели бы попробовать SLED на рабочем столе.**

Network Appliance Inc, Novell Inc.

Интервью

Нат: громкие слова

«Впервые Linux абсолютно готов». Гуру по настольным системам Linux от Novell объясняет, почему он в этом так уверен.

Фото Ната Фридмена и с LinuxWorld: David Blankenhorn

Нат Фридмен [Nat Friedman] – один из всемирно известных хакеров Linux: вместе с Мигелем де Икаса [Miguel de Icaza] он основал Ximian в 1999 году, чтобы разрабатывать и углублять ПО Gnome, и после приобретения Ximian работает на Novell. Работа Фридмена как вице-президента по разработке настольных систем Linux – убедиться, что Linux с точки зрения этих систем годится для бизнеса. Мы отловили его, чтобы узнать его взгляды на Linux на предприятии.

Linux Format: Если компания развертывает продукты Novell, обязательно ли это должен быть «стек» SLES и SLED? Можете вы представить ситуацию, в которой компания имеет, скажем, SLES на своих серверах, но Red Hat или Windows на рабочих столах?

Нат Фридмен: Ну, вы описываете ситуацию, которую мы сейчас имеем, когда у нас есть клиенты Windows и серверы Solaris, Linux, мейнфреймы и прочие. Я думаю, нас ждет неоднородность: Linux прививается, но организация со 100% настольных систем Windows не в один миг придет к 100% SUSE Linux. Но в первый год они могли бы определить долю пользователей, для которых подходит рабочий стол Linux. Мы назвали их «базовыми офисными работниками» – это люди, которые живут в электронной почте, web-браузере, текстовом редакторе, электронной таблице, и которым нужны только базовые функции рабочего стола.

Теперь, со SLED 10, мы можем говорить, что Linux впервые совершенно готов для этих людей; для них больше нет барьеров по функциональности или удобству. Раньше мы этого никогда не говорили, это чрезвычайно сильное заявление, но мы на том стоим. Есть еще люди, для которых он не готов: те, кому ежедневно нужны *Microsoft Project* и Visio и десятки других приложений и кто считает себя «знающими работниками». Люди, разбирающиеся в ПО, установленным на их машине. А для людей, использующих машины, управляемые корпоративно, Linux абсолютно готов, и предприятия могут рассмотреть перевод до 40% в первый год, 70% в следующий; но мгновенного перехода от Windows к Linux не будет.

LXF: Могут SLED и SLES совмещать долгосрочную поддержку и стабильность с потребностью в добавлении новых функций, драйверов и обновлений ПО?

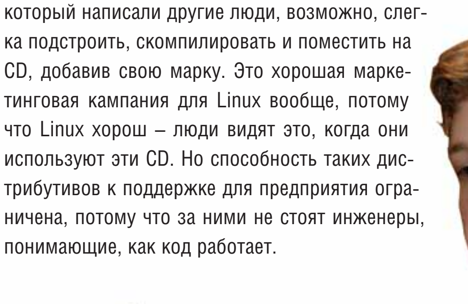
НФ: Сейчас это постоянно муссируется Linux-сообществом: люди приносят новые функции, интерфейсы пользователей и новые идеи в Linux. Некоторые делают это способом, не разрушительным по отношению к текущим пользователям, а некоторые – разрушительным. Можно, например, так изменить API ядра, что аудиодрайверы перестанут работать. Это деструктивно. Или вы вносите новое приложение, не требующее изменений в нижележащих библиотеках, и мы можем его просто подхватить.

Есть проект под названием *Diva*, видео-редактор – мы не поставляем его в SLED 10, потому что он еще не готов, но мы могли бы выбрать его для добавления впоследствии. Вот тот тип изменений, которые мы можем делать. Я думаю, новые версии Gnome или KDE не имеют особого значения по большому счету; большинство проблем и сомнений, с которыми люди сталкиваются в настольных системах, это не проблемы оконного менеджера, или файлового менеджера, или панели. Когда я общаюсь с предприятиями, основные проблемы, которые мы видим – интернет-страницы, которые не работают должным образом в *Firefox*, потому что они используют специфичный для Microsoft DHTML или ActiveX, или «чистые» Windows-приложения.

LXF: Как вы относитесь к выходу Ubuntu на корпоративный рынок с их релизом 6.06 с долгосрочной поддержкой?

НФ: Во-первых, я сказал бы, что не считаю доверие корпораций и причастность к сообществу взаимоисключающими вещами. Я думаю, как раз это мы в Novell и демонстрируем все больше и больше, каждый день. Говоря конкретно об Ubuntu, одно из преимуществ, которое мы имеем сейчас в Novell перед любым другим Linux-дистрибутивом, это наша способность поддерживать весь дистрибутив – полностью, от ядра до электронной таблицы. Так что если вы придете к нам с проблемой ядра, допустим, по его производительности, у нас есть инженеры – эксперты по ядру. Мы, вероятно, имеем самую высокую концентрацию экспертов ядра, и можем исправить эту проблему. Если электронная таблица не работает так, как вы хотите, мы сможем исправить и это.

По-моему, ряд Linux-дистрибутивов имеет привычку загрузить открытый код, который написали другие люди, возможно, слегка подстроить, скомпилировать и поместить на CD, добавив свою марку. Это хорошая маркетинговая кампания для Linux вообще, потому что Linux хорош – люди видят это, когда они используют эти CD. Но способность таких дистрибутивов к поддержке для предприятия ограничена, потому что за ними не стоят инженеры, понимающие, как код работает.







Узнайте больше!

Чтобы узнать больше от Ната, включая его мысли о Mono 1.2, посетите www.linuxformat.co.uk/mag/nat.html.

Будущее под обстрелом

Куда Novell пойдет дальше?

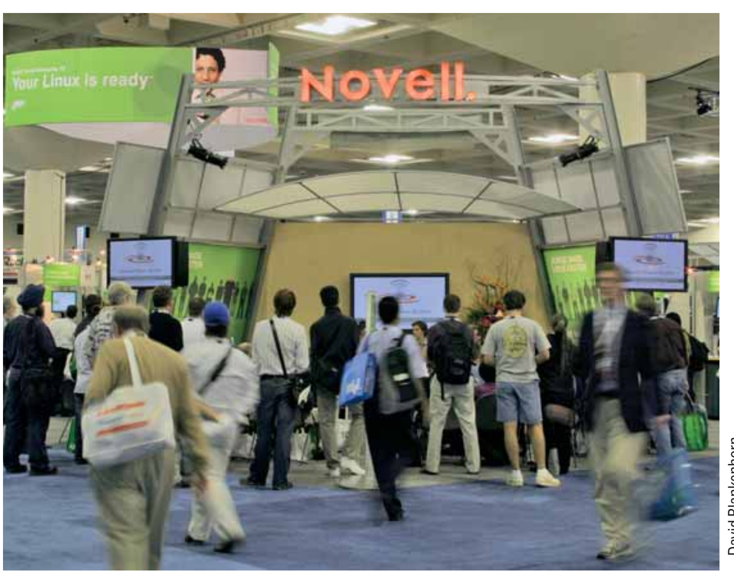
На LinuxWorld в Сан-Франциско в середине августа, Джон Драгун [John Dragoon], директор Novell по маркетингу, сообщил, что было свыше 325 тыс. загрузок SUSE Linux Enterprise 10 за первый месяц его существования. Это серьезная, впечатляющая статистика. Даже если лишь небольшая часть этих пользователей в конце концов заплатят за лицензию, Novell может занять очень сильную позицию. Еще более впечатляет, что SLED получил на LinuxWorld награды Best Of Show и Best Desktop Solution. Нет сомнений, что к корпоративным предложениям Novell относятся серьезно, а компания становится все более и более открытой для сообщества. Наличие таких ударных хакеров, как Мигель де Икаса, Майкл Микс [Michael Meeks], Нат Фридмен и Грег Кроа-Хартман, помогает ей в обоих отношениях.

Цикл разработки также выглядит стабильным. SLES 9 был выпущен в августе 2004 года, со SLES 10, выпущенным почти 24 месяца спустя. Мы должны увидеть SLES 11 в 2008 году – есть надежда, что после предыдущего релиза пройдут те же 24 месяца. Множеству новых технологий, с которыми мы экспериментируем сейчас, это позволит достичь уровня зрелости, пригодного для рынка серверов. Планы по SLES 11 включают виртуализацию для серверов и хранилищ и добавление управления профилями в *AppArmor*, а настольная версия будет предлагать полностью интегрированную поисковую машину *Beagle*. NetWare будет все еще доступна, но она будет паравиртуализирована в виртуальную машину, работающую на SLES. Novell, возможно, в прошлом не была уверена в том, как использовать SUSE по максимуму, но похоже, теперь почва для него найдена.

> «Планы включают виртуализацию для серверов и хранилищ.»

И это благодаря открытой модели. Novell сумела трансформироваться из фирмы, работающей исключительно с проприетарным ПО, в фирму, осуществляющую огромные вложения в разработку открытого кода. И хотя сначала эта стратегия управлялась финансами, превращение в Open Source-предприятие добавило Novell шарма и принесло существенное число новых клиентов, впервые после выпуска NetWare. Джон Пелан [John Pelan], ведущий системный консультант Gatsby Computational Neuroscience Unit, поясняет: «Нам нравится лицензирование Open Source, не из фанатизма, а по практическим соображениям. Самое существенное из них – вы имеете полный контроль над вашими основными сервисами. Если у вас ПО с проприетарными лицензиями, то когда что-то разладится, вы оказываетесь во власти разработчика ПО. Прошлый опыт подтверждает это.»

Д-р Пилен заставляет SLES работать в конфигурации с высокой доступностью, использующей 16 ядер Opteron и кластеризованную файловую систему, основанную на второй версии Oracle Cluster File System. Neuroscience Unit пытается исследовать протекание мыслительных процессов, используя теоретические модели, разработанные с помощью компьютеров. Открытая разработка изменила фирмы, подобные Novell, и это меняет наше окружение. LXF



David Blankenhorn

› **Технологию от Novell тепло встретили этим летом на LinuxWorld, где компания анонсировала некоторые новые выгоды использования SUSE Linux Enterprise.**

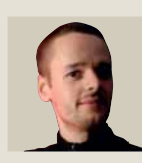
Глядя с колокольни LXF



Грэм считает:
«Изначально, я бы сказал, что Novell недооценила открытую природу Linux. Ей также недоставало истории, чтобы суметь построить увлеченное открытое сообщество вокруг своих корпоративных дистрибутивов, как это сделал Red Hat. Но в последнее время все, похоже, меняется. SUSE становится все сильнее и сильнее, и ее релиз SLES был хорошо принят. Если Novell просто откроет ряд своих основных технологий, особенно некоторые из старых сервисов каталогов или даже код NetWare, она завоюет большее доверие сообщества и добьется сильной позиции к тому времени, как выйдет SLES 11.»



Пол считает:
«Novell пугает меня, и я не верю ей как член сообщества свободного ПО. С одной стороны, нанимает несколько дюжин первоклассных хакеров для работы над крутыми проектами типа Mono. А с другой стороны, прошло всего шесть месяцев с тех пор, как Novell сказала «Делайте правильный выбор: выбирайте SUSE Linux вместо Red Hat». Не для того ли существует Linux, чтобы не зависеть от поставщиков? Вряд ли для того, чтобы наши поставщики грызлись между собой, чтобы оттяпать друг у друга львиную долю 5%-ного сектора рынка, пока Microsoft гребет прибыль миллиардами долларов. Хватит болтать о рынке, Novell; стань командным игроком, и все мы будем в выигрыше.»



Майк считает:
«Важнее всего то, что возрастающая активность Novell на рынке корпоративных Linux обеспечивает выбор. Red Hat – хороший член сообщества, но важно иметь как минимум двух основных поставщиков Linux-дистрибутивов: это стимулирует конкуренцию и дает потребителям выбор источника ПО, недоступный для других ОС. Хотите Windows? Ступайте в Microsoft. Хотите Linux? Выбирайте компанию, которая вам лучше всего подходит. Novell все еще рассматривается многими как «одна их тех старых фирм по серверам», но с учетом количества приобретенных ею талантливых Linux-разработчиков, этот имидж постепенно преображается.»

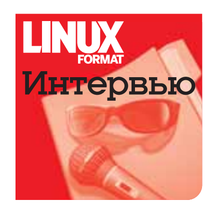


Ник считает:
«Ясно, что Novell приобрела SUSE в надежде оживить ее предложение и построить будущее для ее программ и услуг. Подход может отличаться от Red Hat, и возможно, отличаться от целей других организаций, играющих на пространстве Linux, но Novell внес фантастический вклад в сообщество, в основном финансированием разработки и совершенствования проектов типа Mono, *OpenOffice.org*, *Apache*, Eclipse, GCC, Perl... список можно продолжать бесконечно. Обычно это связано с вещами, которые Novell собирается выпустить (например, рабочий стол для офиса) – именно так большой бизнес и должен поддерживать Open Source.»

› Как вы оцениваете Linux-предприятие Novell? Сообщите нам, как вы думаете, Novell работает в бизнесе или в сообществе, на **lxf.letters@futurenet.co.uk**.

Новый старт

Еще одна настольная ОС, но не являющаяся очередным дистрибутивом Linux? Linux Format беседует с **Кристианом Ван Дер Влиетом** – человеком, который стоит за интригующим проектом по продвижению свободного ПО в массы...



Syllable – это попытка привести ОС с открытым исходным кодом на домашние компьютеры, не используя ядро Linux. Его разработчики с самого начала поставили целью обойтись без останков Unix'а семидесятых и структуры, центром которой является разработчик и которая, возможно, тормозит развитие Linux. Команда Syllable (www.syllable.org) работает над созданием собственного ядра, интерфейса пользователя и приложений, стараясь, чтобы они летали с быстротой молнии даже на старом оборудовании.

Проживающий в Бристоле Кристиан Ван Дер Влиет отвечает за разработку Syllable, координацию работы программистов по всему миру и общее направление проекта. Он не ленив критиковать Linux, но и не считает, что Syllable уже завтра изменит мир: у него свои взгляды на программы с открытым исходным кодом для настольных ПК и свой план реализации этих взглядов. **Майк Сондерс** [Mike Saunders] пригласил его в Бат: пусть почувствует старомодное сомерсетское гостеприимство.

Linux Format: Расскажите, что такое Syllable, что в нем особенного?

Кристиан Ван Дер Влиет (КвдВ): Это ОС для настольных ПК, и мы особо подчеркиваем, что она рассчитана на малые предприятия и на домашнего пользователя. С самого начала все делалось так, чтобы подходить этому домашнему пользователю. У Syllable собственное ядро, собственная архитектура драйверов, собственный графический пользовательский интерфейс и библиотеки. Кроме того, у нее собственные приложения – вы сможете прочитать файл через HTTP так, как будто он хранится у вас на жестком диске.

LXF: Как с ioslaves от KDE?

КвдВ: Да. Мы стараемся сделать API как можно более ортогональным для разработчиков программ; у нас есть штука под названием 'streamable I/O classes' – это абстрактный класс, который можно читать и куда можно писать вне зависимости от того, соединен ли он с файлом, потоком HTTP или с FTP. Он прозрачен для приложения.

»

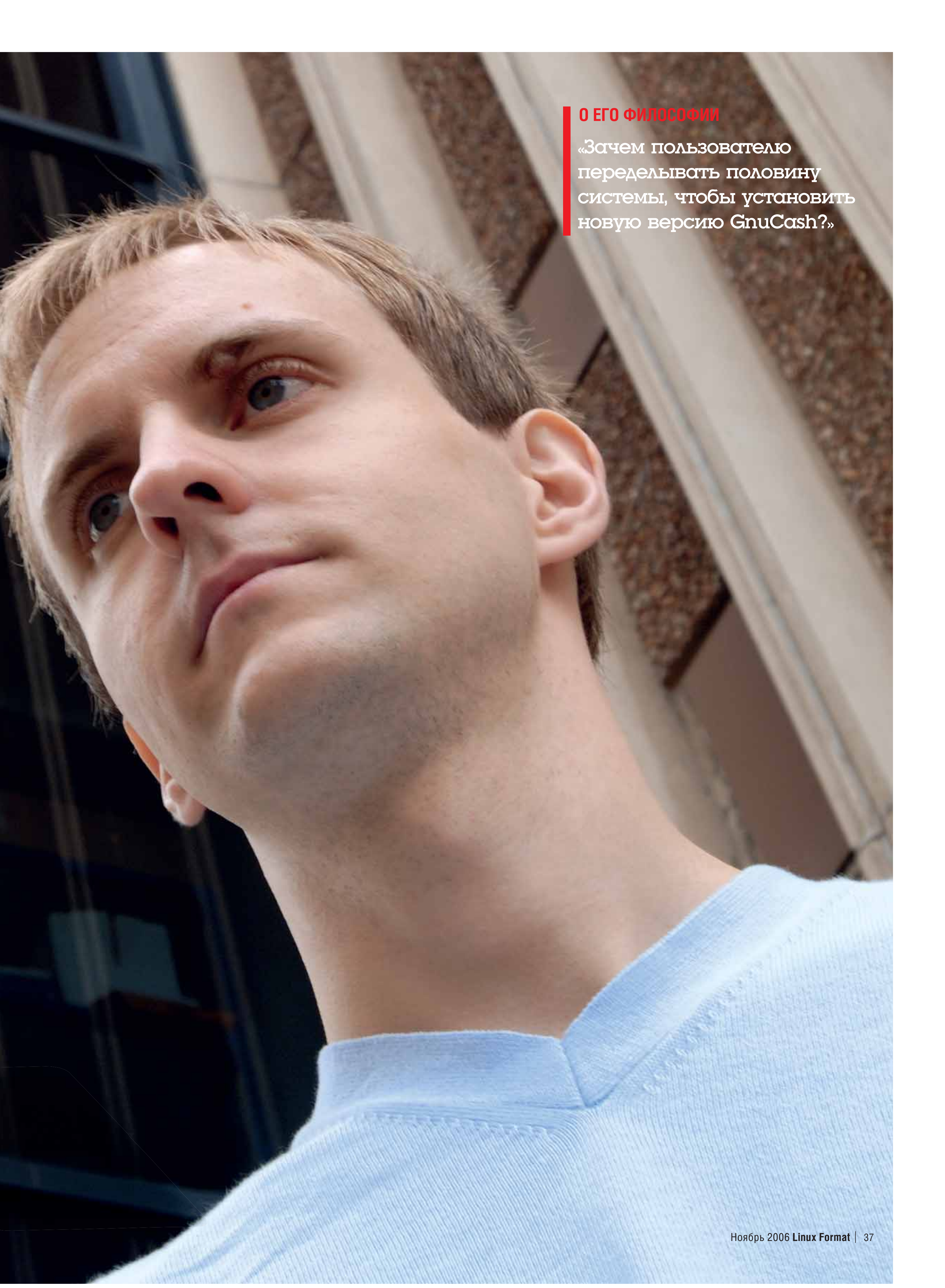
LXF ВИЗИТКА

Кристиан Ван Дер Влиет

» Разработчик операционных систем, создавший Syllable в продолжение AtheOS. Сейчас управляет проектом и руководит командой разработчиков Syllable. Днем работает в телекоммуникационной фирме в Бристоле.

Возраст	27
Национальность	Британец
Работает в Linux	9 лет
Языки программирования	4
Количество ПК	7
Ежедневная норма кофе	0 чашек
Пар сандалий	0

Top Trumps courtesy: Winning Moves UK LTD



О ЕГО ФИЛОСОФИИ

«Зачем пользователю переделывать половину системы, чтобы установить новую версию GnuCash?»

LXF: Многие могут сказать, что Linux начинает играть роль ОС для дома и малого офиса. В чем отличие Syllable?

КвдВ: Нет такой операционной системы – Linux: есть разные дистрибутивы, использующие Linux в качестве ядра. Это коллекции разных компонентов, надерганных отовсюду: ядро Linux, программы GNU, *glibc*, *X.org*, разнообразные наборы инструментов, *GTK*, *Qt*, а поверх – рабочий стол. А потом еще со всех сторон появляются сторонние компоненты и приложения.

В Syllable все появляется вместе, и поскольку это – цельная ОС, у нас есть все, от приложений до ядра. Если мы решаем что-то сделать, мы не спрашиваемся у множества разных групп. Например, в самом последнем релизе Арно Кленке [Arno Klenke, плодовитый разработчик Syllable] реализовал новую систему отрисовки – очень похожую на EXA, горячую новинку в *X.org*. Добавление подобной функции в Linux означало бы необходимость переписать всю архитектуру драйверов и особую работу с X-сервером. А в Syllable мы ее просто добавили.

LXF: Влияет ли этот стиль разработки на конечного пользователя?

КвдВ: Так проще достичь консенсуса среди разработчиков. Не требуется обращаться к разным группам разработчиков и умолять их разрешить нам изменить работу ядра. Если нам нужно, чтобы новая система обращалась к ядру, мы просто делаем это. И проще узнать, кто чем занимается, и никто не наступает друг другу на ногу.

LXF: Вы считаете, что даже с FreeDesktop.org и иконками Tango Linux навсегда останется сборищем разбросанных проектов?

КвдВ: Да, такова ментальность, ее не изменишь. Над всей ОС нет главной фигуры, и если кто-то пытается принять решение, всегда найдется и тот, кому оно не понравится. А если кому-то не нравится то, что вы делаете – ответвляйтесь!

ОБ ОГРАНИЧЕНИЯХ LINUX

«По-моему, тщетно пытаться превратить Linux в настольную ОС.»

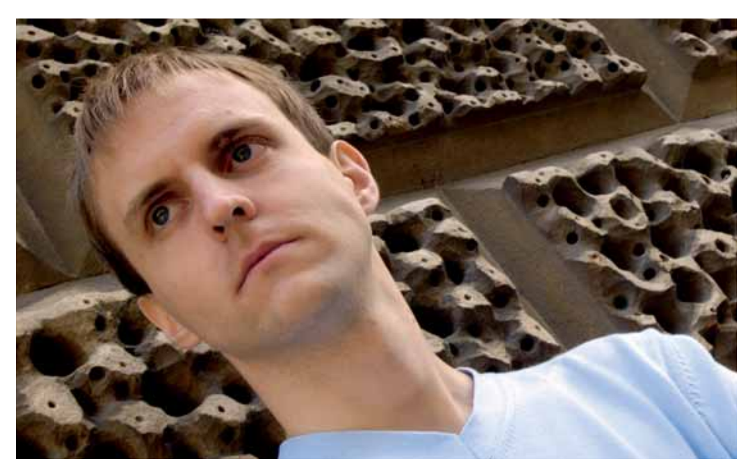
LXF: А если бы Syllable пришлось разветвиться?

КвдВ: В Syllable ситуация отличается тем, что это – цельная ОС, а не составленная из отдельных модулей. Если вы решите сделать ответвление Syllable, вы все равно будете использовать тот же сервер приложений [GUI] и ту же файловую систему. Это не Linux, где вы можете ответвить дистрибутив и заменить рабочий стол на что попало; Syllable – вертикально интегрированная платформа. Мы не пытаемся спрятать низкоуровневые библиотеки за другими библиотеками. В Linux ввели HAL-уровень, пытаясь развязать оборудование и рабочий стол, но Syllable этого не надо, потому что в Syllable вы работаете на рабочем столе Syllable. Нам не надо беспокоиться о версиях библиотек C, версиях ядра и т.п.

LXF: Тогда как вы объясните, что Syllable пользуется «сторонними» проектами, типа *glibc* и линейки инструментов GNU?

КвдВ: Давайте возьмем последний пример с CUPS, когда я пытался реализовать печать в Syllable. Я портировал CUPS, драйвера Gutenprint и Ghostscript. Оказалось, что все это слиплось с *Foomatic*, то есть с большим скриптом Perl, который прячет спулер, будь то CUPS или более старый LPD. В Syllable это совершенно излишне – нам этого не надо. Но нам нужен файл конфигурации PPD, чтобы заставить работать принтер, а все эти файлы заточены под *Foomatic*.

Так что периодически мы сталкиваемся с проблемой: предполагается, что все работает так же, как Linux, и это – препятствие. Другие проекты [неявно] подразумевают вещи, выглядящие в отличных от Linux системах (том же Syllable) несколько по-другому.



LXF: С точки зрения структуры системы, почему в ядре Syllable установка драйверов проще, чем в Linux?

КвдВ: Ядро Syllable обладает очень специфичной функцией: в нем есть ELF-загрузчик в ядре, соответствующий **ld.so** в Linux. То есть нам не нужен специальный загрузчик в пространстве пользователя для исполняемых программ – это делает само ядро! Драйвера в Syllable – это разделяемые ELF-объекты, которые загружаются напрямую, и у нас стабильный ABI [application binary interface].

LXF: В Linux, если вы используете драйвера третьей стороны и обновляете ядро, вам надо пересобирать эти драйвера – и это сложно.

КвдВ: Мы несколько более прагматичны. Линус сказал, что ему не нужен стабильный ABI для драйверов, потому что он не хочет поощрять драйвера с закрытым исходным кодом. Цель благородная, и в этой точке зрения есть смысл. Однако для пользователей и для разработчиков это создает проблему: разработчики оборудования не всегда хотят создавать драйвера с открытым кодом, особенно для видеокарт и беспроводного оборудования, потому что тем самым они раздают свою интеллектуальную собственность. Если пользователь должен выбирать между на 90% закрытым Windows, но с набором полезных открытых приложений, и Linux, где с открытым кодом почти все, но оборудование не работает, то выберет он решение с закрытым кодом, и это не здорово. Вот и выходит, что, имея целью привести людей к использованию открытого ПО, многие разработчики Linux фактически толкают пользователей к ОС с закрытым кодом, а хуже и быть не может. Но я не стану – никогда! – поставлять с Syllable драйвера с закрытым кодом. А если когда-нибудь Syllable станет настолько солидной, что компании типа Nvidia или Broadcom решат создать для нее закрытый драйвер, я не против. У нас есть стабильный ABI, а драйвер – всего лишь файл, его надо скопировать в нужное место, возможно, перегрузиться – и все. Это – часть курса на простоту использования.

LXF: А как насчет приложений?

КвдВ: Распаковываете архив в папку **Applications** – и все. Это цель дизайна системы. Хакеры любят играть с компьютерами, а пользователю просто надо, чтобы все получалось. Им не надо мучиться по поводу какой-то там установки и тратить часы на то, чтобы заставить какую-то программу работать. Зачем им обновлять половину системы, чтобы установить, допустим, новую версию *GnuCash*?

LXF: А если появятся программы для Syllable, зависящие не от библиотек Syllable, а от какой-нибудь другой ерунды? Не получите ли вы тогда тот же ад зависимостей, что и Linux?

КвдВ: Мы практикуем совершенно иной взгляд на зависимости. Мы вообще не занимаемся их управлением. Узнав об этом, люди холодеют от ужаса; но у нас в ходу такие вещи, как статическая компоновка. Слишком часто бывает, что разработчик открытого ПО берет небольшой кусочек кода и создает из него разделяемую библиотеку, но на самом деле это никому не нужно, пока библиотека не начнет использоваться проектах в пяти, а то и больше. Так что в Syllable мы спрашиваем разработчика: вы действительно будете обновлять эту библиотеку независимо от приложений? А еще в Syllable такие ресурсы, как значки, упакованы в само приложение, а не разбросаны по всем директориям.

LXF: Как вы занялись программированием?

КвдВ: Я начинал на Commodore 16, который многие в глаза не видели, или даже и не слышали о нем. Он был бедным родственником C64, только интерпретатор Basic у него, как ни странно, был получше. Потом у меня появился Spectrum +2A, я на нем программировал в Basic (плохо), а потом у меня был Amiga 500, на котором я вообще не программировал. Сейчас, оглядываясь на то время, я сожалею, что не изучил тогда Ассемблер, что не начал все делать быстрее. Мне было не меньше 18 лет, когда я начал ваять в C. Я начал учиться в колледже, и понял, что мне нужен ПК. Это разбило мое сердце! Помню, я стоял в Waterstones, смотрел на все эти кни-



› Если хотите заниматься разработкой ОС – Вандерс набирает хакеров…

ги, и понимал, что мне надо иметь ПК. Этот ПК у меня до сих пор сохранился – ну, материнская плата и процессор.

LXF: Как вы думаете, то, что вы работали в разных ОС, повлияло на стиль вашей работы в Syllable?

КвдВ: Да – когда у меня появился тот самый ПК, на нем была Windows 95, и я раньше работал в Windows 3.1, NT, и со всем прочим. Любой, кто работал с Amiga, или какой-нибудь другой ОС, не Windows, скажет вам, что Windows 95 была дрянной. К журналу *Amiga Format* как-то прилагался диск NetBSD 0.9 вместе с руководством, и я установил ее на своем A1200. Впечатление она на меня произвела хорошее, но поскольку я тогда не разбирался в Unix, я не знал, как ею пользоваться! А потом у меня появился Amiga 4000, и я установил на него Red Hat 5.1. Так что еще до того, как у меня появился ПК, я уже пользовался Linux, и был весьма впечатлен, а на мой ПК я установил Red Hat 5.2 – это стало естественным шагом к изучению C.

LXF: AtheOS была, так сказать, исходной кодовой базой Syllable, но основной разработчик ее забросил, и Syllable выросла из нее. Когда вы заинтересовались AtheOS?

КвдВ: Я установил Windows 95, меня это не порадовало, и я установил Red Hat 5.2. Но она была сыровата – в те дни, чтобы выйти в Интернет, приходилось писать собственные CHAP-скрипты! Чтобы моя звуковая карта заработала, мне приходилось платить за драйвера, потому что драйвера с открытым кодом не работали. Linux определенно обладал хорошим потенциалом, но он был очень сырым, и на фоне Amiga [как я считал] явно проигрывал. То, что я говорю, многим не понравится, но в 96-м или 97-м так и было.

Я считал, что Linux все-таки лучше, чем Windows, он бесплатный, и можно программировать в нем. Однако по мере его развития в течение пары лет он стал действовать мне на нервы, просто раздражать. Пытаешься что-то обновить – ломается приложение, на которое делалась ставка. А эти зависимости RPM – ведь не было ни Yum, ни Up2date, все приходилось учитывать самому. Помню, как я пытался заставить X работать, и что ни день имела место какая-нибудь ерунда, доводившая меня до бешенства.

LXF: Вы, видимо, уж очень обозлились. Не так много на свете людей, способных разозлиться до такой степени, чтоб написать собственную ОС!

КвдВ: Был один парень в IRC, который сказал: «Я напишу собственную операционную систему». Мне понравилось, как это звучит, и я тоже начал писать свою, с оригинальным названием FreeOS. Это был гибрид Linux и AmigaOS, и я писал планы, как мне работать с библиотеками, файловыми системами, драйверами и всем прочим. Упор был на Amiga, но бралось и лучшее из Linux. А потом этот парень из IRC спросил: «Ты видел AtheOS?» Я загрузил ее, запустил, и меня просто потрясло, насколько хорошо была увязана эта система – как далеко вперед ушел Курт Скауэн [Kurt Skauen, ведущий разработчик AtheOS]. Это была версия 0.1.3, но она была настолько близка к структуре ОС, которая уже была у меня, как будто он прочел мои мысли.

На тот момент в работу над AtheOS было вовлечено два-три человека. У Курта был странный настрой: он решил использовать GPL, но при этом сохранял контроль над всем кодом. Что ж, справедливо: операционная система его, значит, и правила устанавливать ему. Те из нас, кто был включен в список рассылки, присылали драйверы то для одного, то для другого, я написал *AEdit 0.1*.

LXF: А остальное – это уже история… Чем, по-вашему, Syllable отличается от других маленьких ОС, например, ReactOS или Haiku?

КвдВ: На данный момент большинство людей определенно отнесло бы Syllable к классу любительских операционных систем, но у нее уже есть свое большое сообщество. Она находится на стадии зрелости, и ее можно использовать на большинстве компьютеров. В ней реализованы ценные для реального мира функции – аппаратно-ускоренный рендеринг, драйверы для большинства устройств, печать, воспроизведение мультимедиа… Так что она уже на пути превращения из любительской в профессиональную ОС, наверное, так же, как было с Linux в 94 или 95 годах.

LXF: И что нужно для этого превращения?

КвдВ: Нужна коммерческая поддержка. Нужны те, кто захочет поддерживать Syllable, стать людьми для переговоров. Сейчас, если вы при использовании Syllable натолкнулись на ошибку, я попытаюсь исправить ее. Но гарантий на это я вам не могу дать: вдруг в тот момент я буду занят чем-то еще, и найденную вами ошибку исправлять некогда. А может, у разработчиков ядра не хватит времени, чтобы реализовать нужную вам функцию. Чтобы пользователь чувствовал себя уверенно и комфортно, работая в Syllable, необходима поддержка, люди, которые гарантируют разработку в течение последующих двух лет.

В обозримом будущем у меня нет иных намерений, кроме как и дальше программировать для Syllable – я люблю ее, и намерен сделать лучшей операционной системой, насколько сумею. Но если сейчас я предложу использовать Syllable профессионально, меня просто высмеют. Нам нужна способность обеспечить безопасность и гарантию коммерческой поддержки.

LXF: Проект Haiku [см. HotPicks, LXF78], клон BeOS с открытым исходным кодом, недавно сумел на несколько месяцев нанять разработчика – Акселя Дорфлера [Axel Dorfler] – для работы над ОС полный рабочий день. У Syllable есть ресурсы, чтобы сделать то же самое?

КвдВ: Как бы я этого хотел! На данный момент это цель средней дальности: найти человека на поддержку Syllable. Я твердо убежден, что при наличии одного-двух разработчиков, скажем, меня и Арно, работающих над Syllable с полной занятостью, версия 1.0 была бы готова уже года через полтора. Сейчас у нас с Арно достаточно свободного времени для работы над Syllable, но все же мы не можем посвящать этому целый день.

LXF: И какие же приложения вы хотите увидеть в первую очередь?

КвдВ: Все! [Смеется.] Syllable – молодая ОС, и приложений для нее сейчас не так уж много. Мы концентрируем нашу деятельность в двух основных областях: Интернет – и у нас есть такие вещи, как браузер *ABrowse* и клиент Chat Jabber; и мультимедиа – здесь у нас есть видео- и аудиоплейеры. Есть и пробелы, например, нехватает графического FTP-клиента, но все же именно на этих областях мы сконцентрировали наше внимание. По части офиса у нас вообще ничего нет, кроме порта XPDF. Есть порты для Vim и Emacs, но это не «родные» приложения.

LXF: Могут ли сосуществовать Syllable и Linux? Лет через пять, будут ли они обе здравствовать и набирать силу?

КвдВ: Естественно! Я всегда говорил, что сила Linux – это сервер, а Syllable будет дополнять его на настольных ПК. По-моему, тщетно пытаться заставить Linux работать в качестве настольной операционной системы, если Syllable может с этим справиться куда лучше и вернее! LXF

Читайте еще!

Мысли Вандерса о патентах, поддержке Syllable и о наказаниях за неудачные названия программ – читайте на www.linuxformat.co.uk/vanders.html

Что за штука... CSS 3?

Семантическая сеть норовит ободрать стили у Интернета, и **Пол Хадсон** спешит к ордам разработчиков, готовых к ответному удару.



Ну знаю я, что такое CSS: это насчет стилей, типа макияжа web-страниц, верно?

Можно, конечно, и так сказать. Каскадные таблицы стилей (Cascading Style-Sheets), или же способ назначить стили элементам HTML. По крайней мере, все начиналось именно с этого...

Думаете, в CSS 3 будет иначе?

Вообще-то, перемены уже идут. CSS позволяют разработчикам сократить объем кода HTML, дав возможность указать элементам страницы их цвет, используемый шрифт и т.п., не смешивая эту информацию со смысловой частью кода. Когда-то давно, семантические тэги, типа **<h1>** и **<p>**, шли вперемешку с тэгами стиля, типа **<font>** и **<b>**. Семантические тэги описывают структуру документа – определяют части, которые должны быть заголовками, телом документа, вставками и прочее. Тэги стиля описывают стиль документа – например, размер шрифтов и цвета. Если тэг **<font>** использовать для отделения различных частей страницы, то компьютерам не понять, где главный заголовок, где подзаголовок, а где тело документа. Структурные тэги вроде **<h1>** для определения заголовков первого уровня и **<p>** для параграфов помогают компьютерам разобраться в содержании web-страниц.

Надо полагать, именно отсюда пошел термин «семантическая сеть».

Это верно: придать web-страницам смысл, а не стиль.

Как сюда вписывается CSS?

Мы согласились, что CSS – это каскадные таблицы стилей; а точнее – это код, назначающий стили элементам страницы. В семантической сети у самих страниц нет стиля – они содержат только структуру документа. А потом применяется CSS, чтобы описать цвет, шрифт и так далее. Приятно то, что CSS обычно хранится в отдельном от HTML файле, и для переделки дизайна всего сайта – например, если ваша компания затеяла смену униформы – надо будет просто отредактировать один файл.

Значит, CSS даст дизайнерам возможность наводить лоск, а разработчикам маяться с HTML-кодом?

Вы уверены, что так не было и раньше? Конечно, полный контроль дизайнеров над web-страницами вносит определенную долю проблем. Например, многие ранние (да и современные) сайты на основе CSS использовали фиксированные размеры шрифта, ради совершенства дизайна. При мониторе с большим разрешением, плохом зрении или желании, чтобы текст легко читался издалека, приходилось молча страдать – дизайнеры считали, что шрифт хорошо смотрится в выбранном размере, значит, нечего и менять его.

Вот и верь, что CSS делает мир лучше...

Ну, история еще не закончена. Первые разработчики CSS исправлялись и учились использовать мощь CSS для улучшения сайтов, а не дешевого трюкачества. В CSS 2 добавили массу новых возможностей – в частности, селекторы, позволяющие указать, каким элементам нужно задать стиль: например, вы хотите, чтобы все ссылки внутри таблиц были одного цвета, а все остальные ссылки – другого. Там, где CSS 1 определяли высоту и ширину элементов, CSS 2 определяют минимальные и максимальные ширину и высоту элементов.



Итак, мы получаем больший контроль?

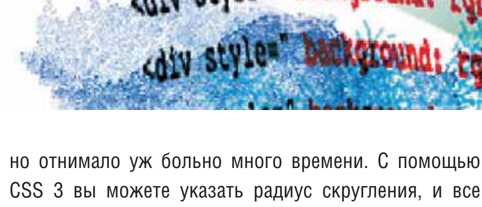
Именно.

А CSS 3 даст еще больший контроль?

Вы получаете кое-какие новые возможности, но это не главное в новой спецификации. Дело в том, что CSS 3 исправляет худшие дефекты предыдущих версий, которые дизайнерам приходилось обходить, используя всякие там танцы с бубном.

Становится уже интересно!

Рассмотрим пример с окантовкой объектов. Как получить скругленные края? Раньше приходилось создавать изображения для углов и накладывать их на элемент, чтобы он хорошо выглядел [в браузерах Mozilla был специальный атрибут CSS, но он не являлся стандартным, – прим. ред.]. Оно, конечно, работало, но отнимало уж больно много времени. С помощью CSS 3 вы можете указать радиус скругления, и все будет сделано за вас.



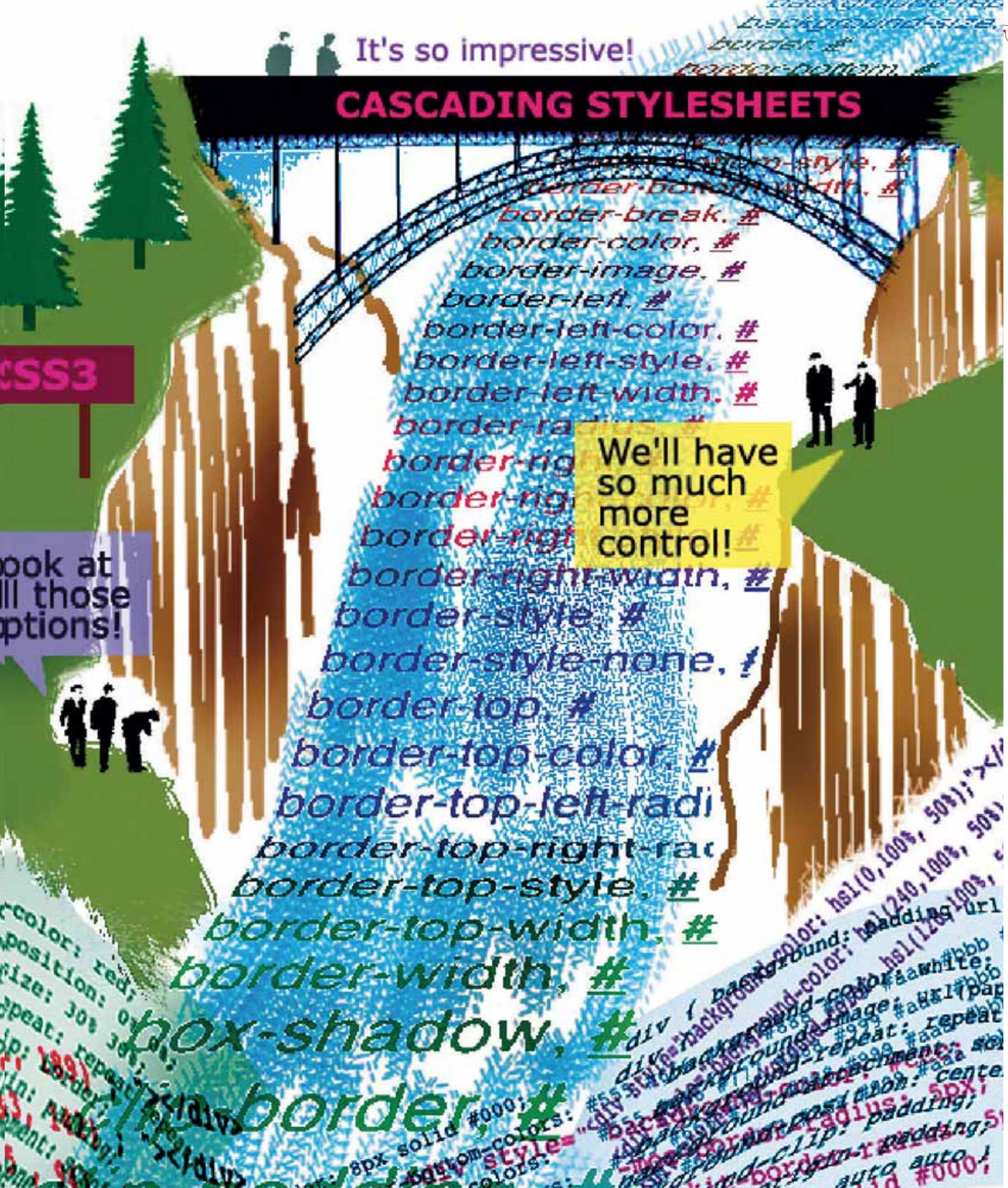
Те целых два парня, что действительно мечтали о закругленной окантовке, будут в экстазе. Как насчет чего-нибудь посерьезнее?

Хорошо, а прозрачность? Раньше это приходилось делать с помощью либо альфа-канала PNG, интенсивно нагружающего CPU (причем без поддержки *IE*), либо чисто Microsoft'овских расширений, использовавших преимущества DirectX. А в CSS 3 для этого появилось два стандартных способа, одобренных W3C.

То вообще не было стандартов много лет, а то вдруг сразу два?

Верно! Первый способ задать прозрачность состоит в задании цвета элемента как RGBA. Это означает Красный, Зеленый, Синий и Альфа, и вы можете определить каждое из первых трех значений в диапазоне от 0 (нет цвета в этом канале) до 255, а альфа определить в диапазоне от 0 до 1 (1 означает совершенно непрозрачный). Если у вашей страницы черный фон и вы помещаете на нее элемент **<div>** со значением цвета 255, 255, 255, 0.5, то он отобразится серым, потому что

Что за штука... CSS3?



смешается в отношении 50% с черным цветом фона. Другой способ состоит в задании атрибута непрозрачности объекта между 0 и 1.

Какие же это стандарты, если их два?

Ну, очевидно, атрибут «непрозрачность» (opacity) надо использовать, когда требуется, чтобы объект просвечивал, но не имел никакого специального цвета. Изображения, например, должны использовать opacity, а не значение RGBA. Они также имеют более тонкое отличие: установка цвета фона элемента **<div>** с помощью RGBA сделает его прозрачным, но не повлияет на объекты, которые он содержит, например, текст. Но если вы используете атрибут opacity, то **<div>** будет прозрачным, как и все объекты внутри него.

Здорово! Почти рыдаю от счастья. Есть еще какие-нибудь чудеса?

Например, вот это: несколько фоновых картинок для любого элемента.

Зачем это нужно?

Ну, положим, вы делаете страничку о мореплавании. Вы хотите, чтобы на вашей страничке фоном были облака, а также, чтобы внизу экрана был океан, слева вверху располагался компас, а справа внизу пиратский флаг. С помощью CSS 2, вам пришлось бы определить несколько элементов и попытаться расположить их, отслеживая прокрутку экрана и изменение размеров. А с CSS 3 – уже догадались? – сделать это гораздо проще. Вы можете теперь определить много различных картинок, а также их положение на экране (абсолютное или относительное), и браузер автоматически расположит их для вас.

Благодаря CSS 3, ленивые разработчики станут еще ленивее?

Нет – все эти примеры говорят о тех вещах, которые делают все в любом случае, но определение их как стандартных позволит а) работать им в любом браузере и б) для их кодирования потребуется намного меньше кода. Возьмите, например, столбцы – мы все читаем текст по колонкам в газетах и журналах (включая этот), а на web-странице мы читаем очень длинные строки текста, ограниченные только размером нашего экрана. До CSS 3 не было возможности создать непрерывный переход текста от одного столбца к другому – вам приходилось создавать все столбцы вручную, затем прикидывать, сколько текста умещается в каждом из них. Начиная с CSS 3, вы можете определить, на сколько столбцов вы хотите разбить элемент, а также размер интервалов между столбцами.

Но сеть-то выглядит намного лучше без столбцов?

Эта статья отвечает на вопрос «что», а не «почему»: я не собираюсь судить о новых свойствах CSS 3! По мне, так больший выбор – это хорошо.

Хватит о примочках для дизайнеров. Вы сказали, что разработчики будут иметь больший контроль с CSS 3 – вот об этом я хочу услышать!

Ладно, поговорим о селекторах. В CSS 3 вы можете выбирать элементы, к которым присоединены определенные атрибуты – строго соответствующие атрибуты (используя ~), соответствия подстрок (с помощью *****), строки, начинающиеся с определенного шаблона (с помощью **^**) или заканчивающиеся определенным шаблоном (с помощью **$**). Так, если вы хотите отобразить зеленым все ссылки на сайты Великобритании, то напишите следующее:

```
a[href$=".co.uk"]{color:green}
```

Новые селекторы полезны также при отладке. Например, подразумевается, что все изображения на ваших страницах содержат альтернативный текст для целей общедоступности, но на большом сайте отследить все изображения довольно трудно. С помощью CSS 3 эта проблема решается следующим образом:

```
img:not([alt]){border:3px solid red;}
```

Все, я подсел на CSS 3 – прямо сейчас за это и возьмусь!

Не гоните коней: CSS 3 – отличная спецификация, но ее надо еще реализовать во всех браузерах.

То есть вы мне тут всякого наговорили, а пользоваться этим нельзя?

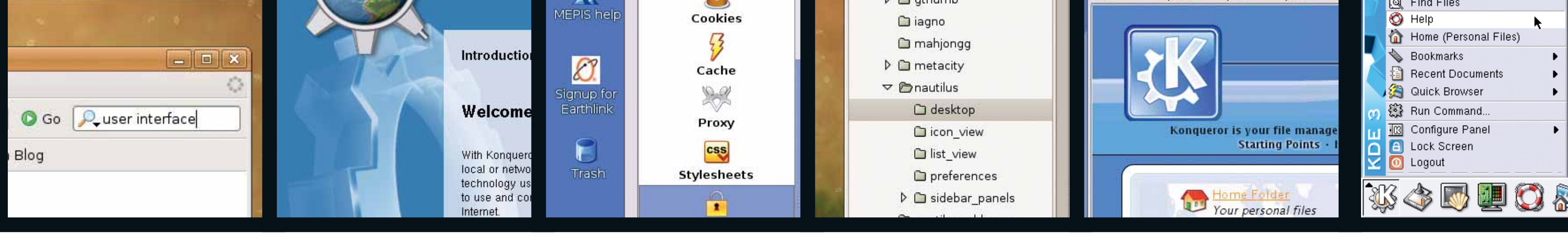
Кое-что уже можно использовать прямо сейчас, но всеобщей поддержки пока нет. Некоторые браузеры поддерживают довольно большой набор, но по крайней мере один (*Internet Explorer*) из них не поддерживает CSS 3 совсем.

Какие браузеры уже используют CSS 3?

Konqueror – на данный момент – возглавляет список, имея полную поддержку всех новых селекторов, а также некоторые другие новые возможности. *Mozilla Firefox* наступает ему на пятки, как и *Opera 9*. Приятно, что довольно большая часть новых функций реализуется в *Internet Explorer 7*, поэтому к моменту выхода Vista все основные браузеры уже будут поддерживать CSS 3.

Журнальная площадь исчерпана, задаю обязательный финальный вопрос: на каких URL можно узнать побольше?

Каких URL? А, хороший вопрос – я ж еще про них не сказал! Официальные спецификации – по адресу www.w3.org/Style/CSS/current-work, полезный сайт с предварительной информацией – www.css3.info, а о том, какие браузеры в каком объеме поддерживают селекторы CSS, можно узнать на странице http://dev.l-c-n.com/CSS3-selectors/browser-support.php. LXF



Проблема с настольным Linux

или Почему я боюсь Gnome 3.0

Не теряют ли рабочие среды направление развития? **Полу Хадсону** видится закат давнего движущего принципа Open Source «выпускать раньше, выпускать чаще»...



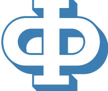
› KDE в SimplyMepis 6.0: хорошие инструменты и системные иконки придают экрану внушительный вид.



› KDE четыре года назад: иконки похуже, но никакой неразберихи, все красиво и быстро грузится.

Факты говорят

Вам важно знать, что здесь все написано не просто со зла. Да, я зол, но на основании фактов, и вы можете их найти по ссылкам в конце статьи (для этого в тексте помещены цифры в квадратных скобках).



Фейерверк – притягательное зрелище, но вы, верно, знаете, что самому его устроить непросто. За энную сумму денег вам дадут в магазине самые крутые ракеты для плохих парней, затем вы вносите их в дом тайком от детей. Когда на улице достаточно стемнеет, вы поджигаете их, отбегаете на безопасное расстояние и ждете… Ждете… Ждете… Обычно секунд через пять ракета издает звук «вжжжж» и плюется огнем или дымом, но может и вовсе не взлететь. Рискнете ли вы снова подпалить ее, с шансами оторвать себе руки, или с горя сложите костер из ваших старых руководств по Unix?

Linux похож на фейерверк – всегда красочный и готовый восхитить весь мир, но почему-то этого не происходит. Установив рядом Red Hat Linux 7.3 и SimplyMepis 6.0 – дистрибутивы, между которыми лежат четыре года разработки тысячами людей – вы почти не увидите разницы, разве что в Mepis иконки покрасивее и есть кое-какие дополнительные возможности.

Конечно, внутри разница между ними огромна: в Mepis вместо ядра 2.4 теперь ядро 2.6, у нас есть HAL, D-BUS, Xen и Mono, и вообще куча всякого добра. Но большинство из этого новичку не очевидно, а каждодневные трудности использования Linux не особо уменьшились. Как недавно сказал один анонимный комментатор: «Главная проблема ALSA в том, что ее настройка – вне понимания простых смертных. Если ваше оборудование само не заработало с ALSA, добиться его работы вам не светит.» **[1]**

Два шага вперед…

RedHat 7.3 – примечательная веха в истории Linux, наводящая на множество интересных сравнений. RH 7.3 поставлялся с *Mozilla 0.9.9*, потому что дистрибутив вышел в мае 2002 – за месяц до выхода финальной версии *Mozilla 1.0*. *Firefox* тогда был сгустком нейронов в головах Блейка Росса [Blake Ross] и Бена Гуджера [Ben Goodger]. Но нам понятно, в чем они видели проблему: Гуджер – человек, стоящий за пользовательским интерфейсом *Firefox*, и он проделал огромную работу по упрощению сложных участков *Mozilla*. Возьмем, к примеру, диалог Preferences: в *Mozilla 0.9.9* приходилось просматривать 7 различных категорий и около 30 подкатегорий настроек. В *Firefox 1.5* все было разделено на 6 категорий и почти ни на одну подкатегорию.

Пока семейство *Mozilla* упрощалось, *Konqueror*, наоборот, опухал от настроек. Открыв меню Settings в *Konqueror* из RH 7.3, вы можете выбирать из Configure shortcuts, Configure Toolbars и Configure Konqueror. В последнем пункте вы получите 11 новых категорий выбора – не идеально, но и не ужасно. В *Konqueror* из Mepis 6.0 теперь есть 5 различных категорий настроек в меню Settings (наш привет Configure Extensions и важнейшему пункту – Configure Spell Checking, воткнутым в произвольные места меню). В самом окне Configure Konqueror у вас 18 различных категорий, причем некоторые воистину ужасны.

Gnome под огнем

Gnome знаменит простотой использования, но тоже не избежал попреков. Два года назад Linux-комментатор Николас Петрели [Nicholas Petreley] сказал, что «к появлению версии 2.6 пользователь был вправе ожидать, что ему не придется адаптироваться к новому сдвигу парадигмы пользовательского интерфейса, хотя бы по основополагающим моментам типа навигации по каталогам. Однако именно этому и придется переучиваться в последней версии Gnome.» **[2]** Он имел в виду новый пространственный режим навигации в *Nautilus*, в котором при открытии каждого каталога создавалось новое окно. Такой режим огорчил многих пользователей, но им пришлось ждать шесть месяцев, пока в релизе Gnome 2.8 ввели опцию его отмены. А до этого они запускали *Gconf*, искали **/apps/naultilus/preferences** и изменяли значение always_use_browser на 'true'. Наглядно? Не надо песен.

Gconf используется как свалка всех опций приложений Gnome, которые разработчики не хотят показывать в программах. При переходе с Gnome 1.4 на 2 было задумано отличиться от Очередного рабочего стола простотой использования:

убрали массу опций, почистили интерфейс, и рабочий стол сделался спартанским, но простым для изучения. Многие опции пользовательского интерфейса перекочевали в *Gconf*, и их надо настраивать вручную. Тем же методом скрывает свои настройки *Firefox* – переход по адресу URL about:config позволяет увидеть все опции, которые можно редактировать вручную.

Поймите меня правильно: *Gconf* – чудесная программа редактирования опций; лучше, пожалуй, и некуда. Но эквивалента в интерфейсе пользователя не имеют столь много опций, что это уже ограничение функциональности. Например, Ubuntu по умолчанию поставляется с чистым рабочим столом: ни иконки Компьютер, ни иконки Домашний каталог, и т.д. А лично мне они нравятся; и как я могу их снова включить? Насколько я могу судить, единственный официальный способ – загрузка Gconf и смена ключа **/apps/Nautilus/desktop/home_icon_visible**. Если вам не лень скачивать и собирать стороннюю программу, можете попробовать *GTweakUI*: ее цель в «предоставлении дополнительных опций настроек Gnome, которые пользователи требуют, начиная с выпуска Gnome 2.0». **[3]**

Раз пользователи требуют эти возможности, почему команда Gnome по большей части их игнорирует? В декабре прошлого года Линус Торвальдс опубликовал в открытой почтовой рассылке OSDL свои соображения по поводу Gnome, сказав: «Подход Gnome «Пользователи не способны разобраться в функциях» – это заболевание. Если вы считаете пользователей идиотами, то одни идиоты к вам и придут... Пожалуйста, просто советуйте людям брать KDE.» **[4]** В следующих публикациях он продолжил: «Если пользователь не может что-то поменять в интерфейсе – дизайн непригоден.» **[5]**

Не движется ли порой Gnome в сторону непригодного дизайна? Самый частый пример не интуитивных вещей – система шаблонов Gnome. Когда вы нажимаете правую кнопку мыши на рабочем столе Gnome, то можете выбрать Создать Документ, а затем взять шаблон из имеющегося списка. Идея-то хорошая, но вот беда: все эти шаблоны вам необходимо создать самому, скопировав файлы в специальный каталог **Templates**, обязанный находиться в вашем домашнем каталоге. Конечно, вам бы хотелось, чтобы у вас был список типов документов, которые программы регистрировали бы в процессе установки, но ребята из Gnome, видимо, считают, что их подход лучше.



› Как и about:config у *Firefox*, *GConf* из Gnome – способ разработчиков «заметать весь настроечный мусор под ковер».

Я бы рад им поверить, но как пользователю догадаться об этом? Установив свежий Ubuntu 6.06, я не нашел пустого каталога **Templates** в домашнем каталоге. Также ничего не дал поиск по ключевым словам 'document' или 'template' в онлайновой справке. В общем, меню Создать Документ – мертвый груз рабочего стола Gnome. А в KDE есть предопределенный список общих шаблонов, и я постоянно использую эту функцию.

Бремя выбора?

Проблема в том, что решение Gnome заставляет пользователей настраивать рабочую среду по своему вкусу, и это противоречит «норме Gnome» (разработчики устанавливают все правильные предпочтения по умолчанию). В принципе, такая же проблема и у KDE: большей части пользователей не нужна особая тонкость настроек и бездна опций. Конечно, по мелочам они что-то и поменяли бы, но не более того.

Дженсен Харрис [Jensen Harris], ведущий программный менеджер группы изучения пользовательского опыта по *Microsoft Office*, держит блог, где регулярно обсуждается теория интерфейсов пользователя. В июне этого года он рассказал об исследовании про панели команд в Microsoft Office 2003 (это гибрид меню/панели инструментов, позволяющий добавлять и удалять стандартные кнопки, создавать свои собственные кнопки и перетаскивать их в любую часть экрана). Каковы результаты? «Программа запускалась с измененной панелью команд не более чем в 2% сессий. Из этих 2% сессий 85% изменяли четыре или менее команд.» **[6]**

Эта информация основана не на слухах или сплетнях, а на данных от доброй сотни миллионов установок *Office 2003*, и довольно убедительно доказывает, что люди куда как комфортно чувствуют себя при настройках по умолчанию. О да, я знаю: те 0,3% пользователей, что капитально меняют настройки (то есть совершают больше четырех переделок) – это целых 300000 человек, и о них тоже надо позаботиться; и я не говорю, что большинство функций мы должны выкинуть ради простоты. Но я считаю, что кнопки, флажки, полосы прокрутки и ниспадающие списки следует использовать экономно. Интерфейсы надо проектировать, а не давать им расти подобно плесени.

> «Большинству пользователей просто не требуются такие возможности по настройке.»

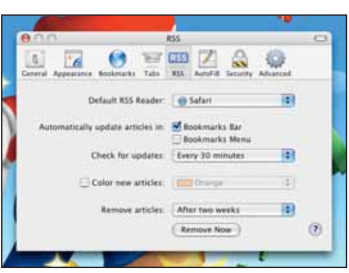
Проект, который, как минимум, пытается найти верный путь, это *Gaim*. Разработчики знали, что их диалог свойств перегружен, поэтому в версии 2.0 удалили большую часть опций и подождали откликов от пользователей. Если поступали жалобы на нехватку опции, ее просто вставляли обратно. Опции, отсутствия которых никто не заметил, были – и правильно! – забыты, потому что люди обнаружили, что им отлично живется и с настройками по умолчанию.

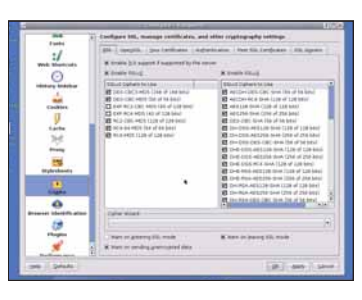
Команда KDE умеет упрощать интерфейс, как мы убедились в 2003 году. Нат Фридмен сказал про KDE, в интервью OSNews, что «диалог настроек у часов содержит аж пять вкладок!» **[7]** Уже через сутки в исходный код KDE поступило исправление, которое реорганизовало и значительно упростило интерфейс.

Кафедральный собор

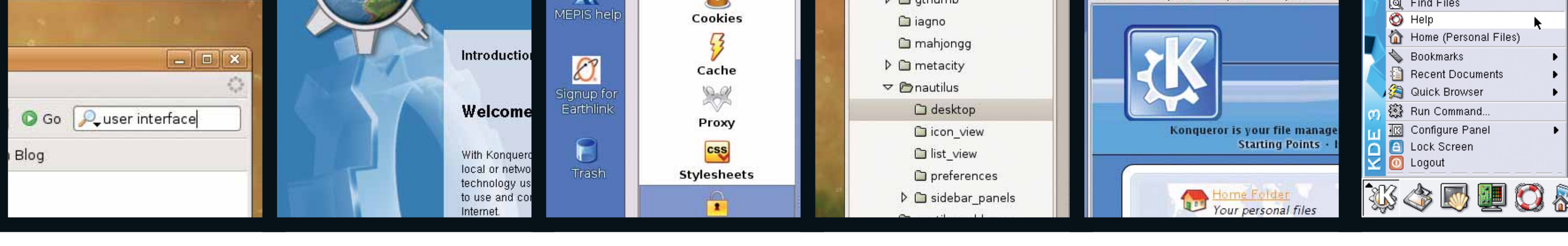
В мире, где есть *Firefox*, *OpenOffice.org*, *Gimp*, *Scribus* и KDE/Gnome, основное отличие дистрибутивов друг от друга – это качество их инструментов настройки и установки: если вы »

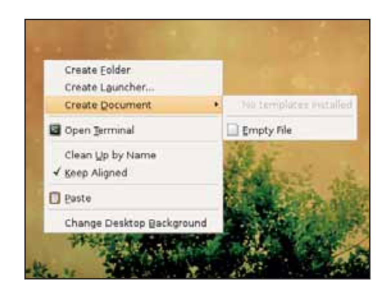




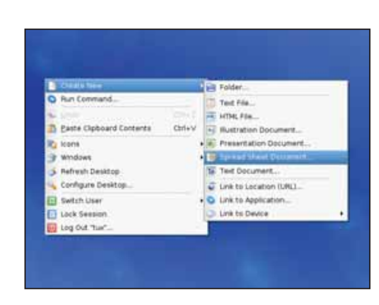


› Диалоги Свойств в *Firefox*, *Safari* и *Konqueror* (сверху вниз) – индикаторы, на сколько каждый проект продвинулся в удобстве и простоте использования.





› Gnome: «Нет установленных шаблонов.» Я: «Почему?» Gnome: «Вот не прочел ты мой исходный код, да и не понял, что нужно создать каталог Templates, болван!»



› Зато KDE позволяет создать новые документы из контекстного меню, используя готовые шаблоны.

» возьмете *YaST* из SUSE, то с точки зрения пользователя он будет не слишком отличаться от инструментов Fedora. А теперь прикиньте: у Ubuntu инструментов настройки почти нет, есть лишь несколько готовых приложений Gnome и пара доморощенных приложений. Однако Ubuntu вот уже два года является дистрибутивом №1 на DistroWatch.

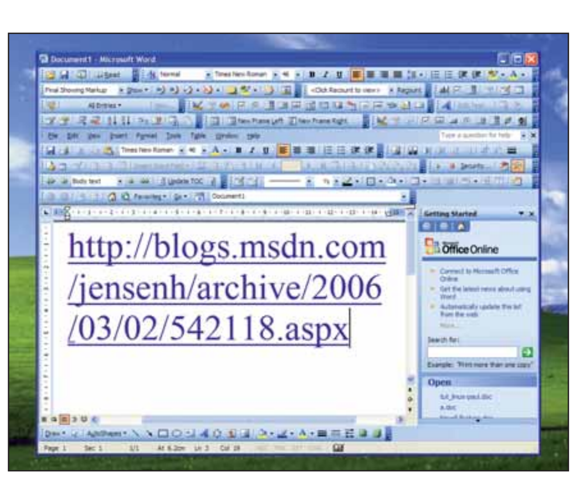
Мать Тереза однажды сказала: «Важно не то, что вы делаете, а сколько любви вы вкладываете в это дело». По-моему, Ubuntu выделяет из общей массы именно желание позаботиться обо всех мелочах, а не запустить грандиозный проект во имя мировой революции. Поставщики Linux это понимали, поэтому эволюционировали постепенно, с шестимесячными промежутками между выпусками: новые версии дистрибутивов выходят два раза в год, а бета- и релиз-кандидаты появляются в промежутках между ними.

Формально, свободное ПО придерживается правила «выпускай раньше, выпускай чаще», означающего «быстро выдать код пользователям и подправить его на основе обратной связи; повторять, пока не будет готова финальная версия».

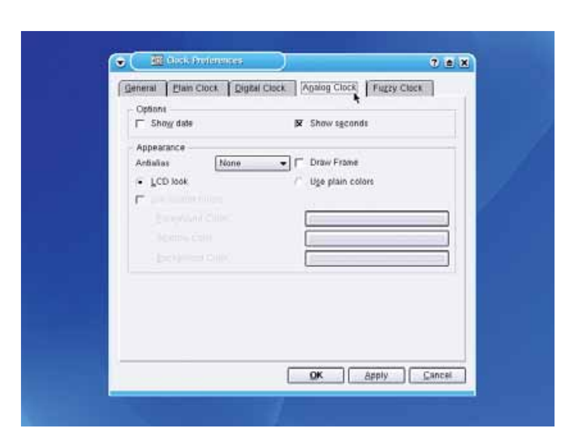
Сравним с Vista: выход Windows XP был в октябре 2001, а Vista вряд ли попадет на прилавок раньше февраля 2007 года, хотя изначально планировался 2003 год, причем будет включать лишь часть заявленных опций, например, Avalon, новый графический движок; Aero, новую графическую тему; Indigo, новую среду коммуникации. Конечно, она запоздает на четыре года и недодаст многие возможности, обещанные Microsoft, например, WinFS.

Такой долгий промежуток времени между выпусками характеризует «соборный» стиль разработки, описанный Эриком Рэймондом [Eric Raymond] в его старом эссе *Собор и базар* [8]. «Соборный» (или «кафедральный») стиль подразумевает выпуск больших порций кода через значительные промежутки времени, что подавляет полезную обратную связь от пользователей. Если люди ненавидят функцию A, то стиль «раньше и чаще» («базарный») позволит отловить это и не дать превратиться в проблему. Но если на функцию A опираются функции B, C, D, и E, то разработчики программного обеспечения должны откатить изменение – а в более реальном сценарии они сделают вид, что проблемы нет, и дождутся, пока всем надоест жаловаться. Сила базарной модели, как и настоящего базара, в открытости: каждый может принести свой товар на рынок. Плохой код немедленно отвергается, а хороший улучшается с момента выпуска.

Теперь позвольте объяснить, почему меня пугают Gnome 3.0 и KDE 4.0: оба проекта переходят на кафедральную модель. После выпуска KDE 2.0 в конце октября 2000 года, средний интервал между смежными релизами 2.x был пять месяцев. Переход от KDE 3.0 к 3.1 занял почти год, а средний интервал между релизами серий KDE 3.x составляет 8,9 месяца. Теперь о KDE 4.0. Уже просматривается 18-месячный цикл разработки KDE 4.0, но можно ожидать, что выпуск финальной версии займет много больше – в частности, потому, что релиз Technology Preview 1 выйдет в конце октября. Спрогнозируйте сами: KDE 3.0 в версии Alpha 1 вышел в сентябре 2001 года, но для выхода финальной версии понадобилось еще шесть месяцев. А когда писалась эта статья, почти три четверти KDE 4.0 имело пометку «Надо сделать» [9].



› По панели инструментов *Microsoft Office 2003* видно: настройка интерфейса может привести к катастрофе.



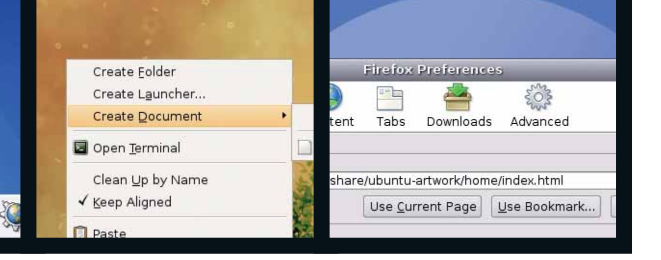
› Прав был Нат Фридман: в старом диалоге Свойства часов KDE действительно имели пять вкладок. Ну и ну!

Учтите, обновление отнюдь не рядовое: новые возможности включают переход на *Qt 4.2*, введение новой графики и новые установки пользовательского интерфейса, портирование на OS X и Windows, добавление нового скриптового движка на основе ECMAScript, замена DCOP на D-BUS, рабочего стола и панелей – на Plasma, переход на новую систему мультимедиа (Phonon), введение слоя аппаратной абстракции (Solid), интегрирование среды коммуникации и добавление контекстного поиска данных (Tenor).

Хотя я всегда приветствую новые возможности, модель разработки KDE 4.0 заставляет вставать дыбом волосы у меня на загривке. Как все эти компоненты поладят друг с другом? Будет ли исправление всех обнаруженных ошибок и внедрение новых функций снова и снова отдалять дату релиза? Как и Vista, KDE 4.0 скорее всего не выйдет в ожидаемый срок конец 2006/начало 2007; и как Vista, он начинает терять обещанные возможности, так как Tenor вряд ли будет готов к релизу 4.0.

Gnome 3.0 также выглядит пугающе, но, возможно, только для меня, потому что я программист и предпочитаю видеть код, а не дискуссию. По мне, «дискуссия» означает «рассуждения о голубизне неба», а уж «рассуждения о голубизне неба» означают «созерцание собственного пупа». Может, это и цинично с моей стороны, так как благодаря тщательной стратегии планирования Gnome обладает цельностью, которая его и прославила, но трудно не поморщиться, читая что-нибудь вроде «а еще мы должны рассмотреть вопрос упрятывания файловой системы» в обсуждении Gnome 3.0 на wiki [10].

Одна из самых интересных козырных идей – переход на рабочий стол, перестраиваемый в зависимости от ваших задач, где вы сами выбираете режимы, отражающие, что вы можете видеть и делать. Идея состоит в создании собственных «проектов», в которых вы определяете, какие программы запустить для определенной задачи, какие контакты требуется видеть, к каким документам нужен быстрый доступ, и так далее. В теории такой подход позволяет уменьшить путаницу и число приложений, с которыми вам надо работать; но по жизни пользователи не заинтересованы в настройке своих рабочих сред. Я думаю, люди работали бы более продуктивно, позволив *Gaim* не показывать контактов, не относящихся к их текущей задаче, но Среднестатистический Джо просто плюнет и отключит *Gaim*.





› Windows Vista: лучше поздно, чем никогда, но даст ли ее появление что-либо новое для пользователей?

Единение как путь вперед

Абстрактные идеи для меня анафема, но кодирование – амброзия, а на сей счет существуют два проекта, которые, как я полагаю, действительно выведут Linux на рабочий стол. Первый – широко обсуждаемый проект унификации рабочего стола, Portland, его цель – предоставить стандартный API для поддержки обычных задач, например, запуска приложения от имени root или открытия любимого пользователем браузера при нажатии на ссылку. Предыдущие попытки [11] далеко не ушли, но Portland уже содержит рабочий код, поддерживается OSDL и принят на вооружение KDE и Gnome. Уже существуют темы, позволяющие выглядеть похоже приложениям *GTK* и *Qt*, а с корректировками Portland они в конце концов станут неразличимы для пользователя.

Вторая важная разработка для Linux – Galago, универсальная настольная среда присутствия. Другими словами, Galago позволяет узнать, кто находится в сети, независимо от того, в каком приложении вы находитесь. Опять-таки, эта возможность уже была не раз реализована, но Galago – первая серьезная попытка сделать такое на всем «пространстве» рабочего стола. Вы, например, читаете письмо от Боба, а ваш почтовый клиент спрашивает у Galago (а тот, в свою очередь, у вашего клиента быстрых сообщений), находится ли Боб в сети, и если да, то над сообщением Боба появится маленькая иконка «в сети», на которую можно нажать и послать ему сообщение.



› Проект Gnome Lowfat демонстрирует, что покажет, к примеру, фотографу рабочий стол на базе концепции задач.

Как видите, оба приведенных мною примера будущих технологий представляют собой интеграцию уже существующих. Я искренне верю, что у нас уже есть все инструменты, необходимые, чтобы Linux стал рабочим столом для всех – чего нам не хватает, это немного любви, чтобы соединить все вместе. Ну не должны вы лезть в справку, чтобы узнать, как работать с диалогом настроек, или недоуменно чесать в затылке из-за того, что меню Создать новый документ пусто; и все же исправление столь элементарных вещей игнорируется ради изобретения новых кодовых имен для технологий, которые по сути делают то же, что ими замещаемые. Суперидеи, обещающие революцию в настольных системах, имеют право на жизнь, но мы достигли нынешней мощи благодаря открытому распространению ПО, которое доказало успех развития систем через эволюцию, а не революцию.

From: Tuomas Kuosmanen <tigert@ximian.com>
To: tigert@ximian.com
Subject: Instant message
Date: Thu, 06 Nov 2003 17:17:53 -0500
Online (AIM)

From: Tuomas Kuosmanen <tigert@ximian.com>
To: tigert@ximian.com
Subject: Instant message
Date: Thu, 06 Nov 2003 17:17:53 -0500
Offline (AIM)

› Информация о присутствии/отсутствии в *Evolution* стоит $2000, если вы сумеете запрограммировать это сами – обратитесь на www.gnome.org/bounties/IM.html.

> «У нас уже есть все инструменты для работы в Linux на настольных системах.»

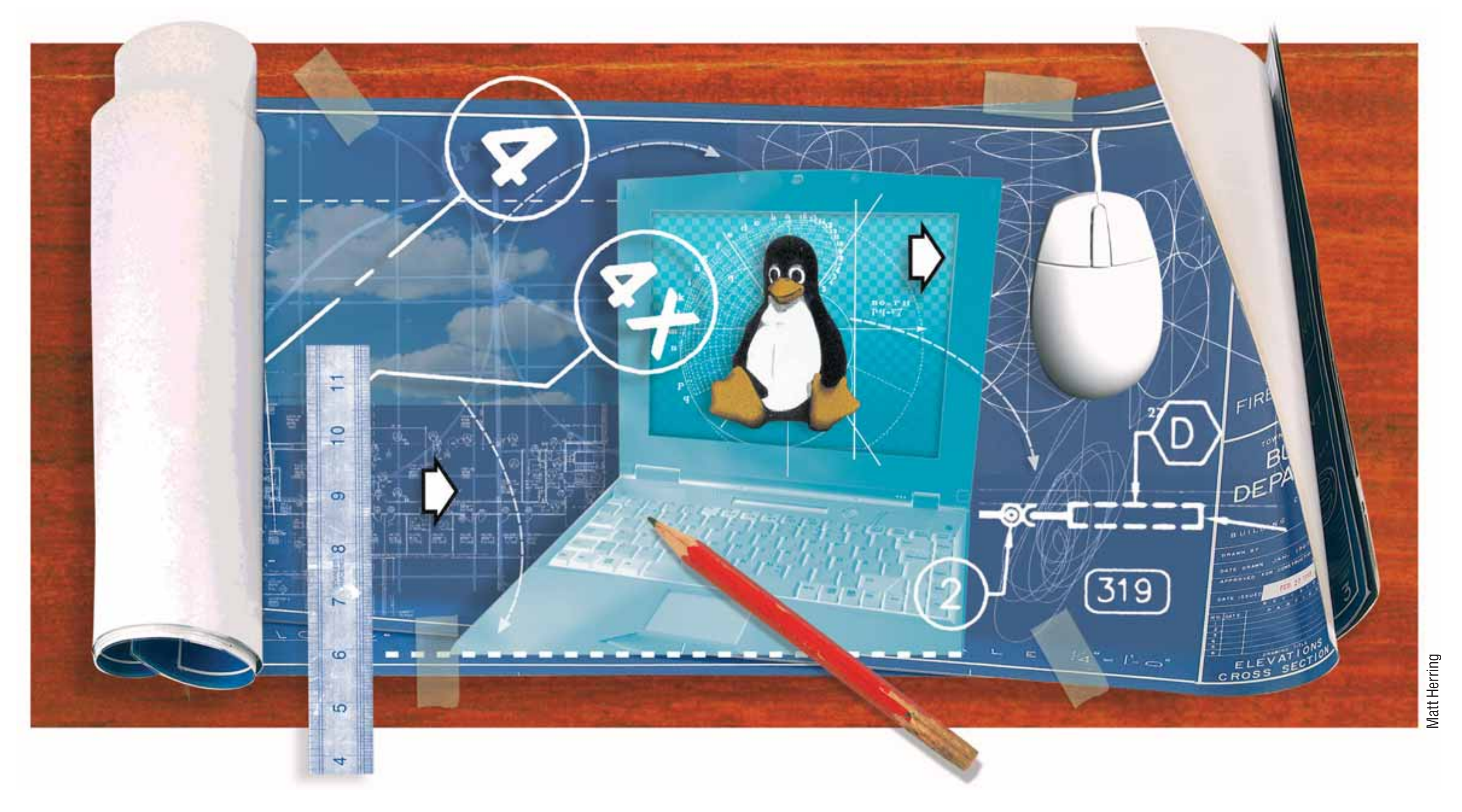
Мы не должны забывать об успехе «базарной» модели. Только опираясь на ее сильные стороны, надстраивая существующие ресурсы и извлекая пользу из огромного числа пользователей, которые азартно тестируют ПО и сообщают об ошибках, можно помочь настольному Linux перестать искриться и сделать то, чего он заслуживает: вспыхнуть, наконец, настоящим фейерверком. LXF

Читателям

Вы считаете, что настольный Linux развивается в неверном направлении, или хотите защитить проект, который вы поддерживаете или участвуете? Согласные и несогласные, пишите на letters@linuxformat.ru

Интересные ссылки

[1] http://chrislord.net/blog/does-alsa-suck.essay
[2] www.computerworld.com/softwaretopics/software/story/0,10801,92934,00.html
[3] http://gtweakui.sourceforge.net
[4] http://lists.osdl.org/pipermail/desktop_architects/2005-December/000390.html
[5] http://lists.osdl.org/pipermail/desktop_architects/2005-December/000395.html
(примечание: готовьтесь увидеть ругательства)
[6] http://blogs.msdn.com/jensenh/archive/2006/06/27/648269.aspx
[7] www.osnews.com/story.php?news_id=3705&page=2
[8] www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar
[9] http://developer.kde.org/development-versions/kde-4.0-features.html
[10] http://live.gnome.org/ThreePointZero



Matt Herring

Ulteo: новая надежда

Что делает Гаэль Дюваль после ухода из Mandriva? Что он думает об изменении мира Linux с момента выхода Mandrake 5.1? Выясняет **Грэм Моррисон**.

На доске почета в редакции LXF одно из лучших мест занимает Гаэль Дюваль, создатель дистрибутива Mandrake, позднее переименованного в Mandriva Linux. Он сделал Linux удобным, похитив отравленную чашу таинственности процедуры установки и несовместимости оборудования и зарыв ее где-то в Редмонде. До Mandrake, Linux был вотчиной исключительно умников. Средневековые карты просто отметили бы территорию Linux как «here be daemons». После выпуска Mandrake почти кто угодно, при некотором желании, мог установить Linux самостоятельно.

Для многих из нас Mandrake был первым дистрибутивом Linux, который вообще работал. При правильной комбинации оборудования (что само по себе дело тонкое), вы могли получить работающий установленный Linux через пару часов после вставки первого CD Mandrake в ваш привод. Простота не возникла случайно: она была прямым результатом большой работы команды разработчиков Mandrake, которая началась с одного человека, занимавшегося этим на досуге – Гаэля Дюваля. Так что когда Дюваль покинул Mandriva в марте этого года и объявил, что берется за новый проект, мы не могли им не заинтересоваться. Когда же оказалось, что этот проект – новый дистрибутив Linux по имени Ulteo, обещающий «изменить подход к использованию ПК», тут уж пришлось перейти к действиям. Это была прекрасная возможность узнать лично от Дюваля, какие процессы вовлечены в создание нового дистрибутива и почему он уверен, что Ulteo будет непохож на других.

Рожденный недовольством

В 1995 Гаэль Дюваль использовал Microsoft Windows 3.1, и, как многие программисты того времени, был недоволен необходимостью платить за среду разработки. Поэтому он поискал дистрибутив Linux, позволивший бы ему создавать собственные приложения и возиться с исходным кодом, и нашел Slackware. После установки с 50 дискет, на которых поставлялся дистрибутив, Дюваль «подсел» на Linux. Потребовалось еще два года, прежде чем Дюваль понял, что Linux в принципе способен заменить Windows. В начале 1998 он перешел на Red Hat и баловался с пробным релизом KDE, который в то время выглядел и вел себя весьма схоже с тогдашней версией Windows (Windows 95), а между делом создавал собственный дистрибутив, приняв Red Hat за основу. Главной причиной выбора Red Hat было его управление пакетами – Дюваль думал о Debian, но счел, что процесс установки слишком мудрен для легкого в использовании Linux. Новый дистрибутив должен был также уметь применять обширную библиотеку RPM-пакетов от Red Hat – подобным преимуществом сегодня обладает Ubuntu, способный использовать большую часть огромного массива пакетов для Debian.

Первая версия Mandrake (5.1) вышла практически одновременно с KDE 1.0, в июле 1998, и это был первый дистрибутив Linux, предоставлявший эту версию KDE прямо из коробки. Дюваль сделал объявление в списке рассылки KDE, обстоятельно разъяснив три главных побудительных причины создания еще одного дистрибутива Linux:

« – Предоставить работающий и легко устанавливаемый дистрибутив Linux людям, которые не хотят тратить много времени на установку и настройку своей системы: просто установи и ПОЛЬЗУЙСЯ.

– Предоставить привлекательную, легкую в использовании систему Linux для новичков, пришедших с другой распространенной ОС, которую вы знаете ;-).

– Предоставить новый дистрибутив в хорошо известной среде Linux (Red Hat 5.1). »

Mandrake взял номер от версии Red Hat, которую Дюваль использовал для создания своего дистрибутива – вот почему первым выпуском Mandrake был 5.1, а не 1.0. Mandrake неуклонно крепнул, впервые доказывая, что Linux может работать и у «простого пользователя». Когда в 2005 году фирма MandrakeSoft, которую Дюваль с партнерами создали вокруг проекта, приобрела разработчика бразильского дистрибутива Conectiva, Mandrake превратился в Mandriva Linux, но сохранил нацеленность на простоту использования. Дистрибутив Дюваля всегда играл важнейшую роль в ознакомлении новых пользователей со свободным ПО, но умел удержать при себе и тех, кто плавал в бассейне Linux без надувного жилета.

И было так...



В начале был Mandrake, и было это хорошо. А затем все изменилось.

Уход Гаэля Дюваля из Mandriva широко освещался и обозначил завершение его участия в созданном им дистрибутиве. Он утверждал, что его уволили; Mandriva объявила, что его сократили из-за реструктуризации, вызванной затягиванием поясов (Дюваль в то время был главой сообщества). Люди тут же захотели узнать, что он станет делать дальше.

Потребовалась всего пара недель, чтобы получить некую форму ответа: пользователей, посещавших персональную страничку Дюваля ради новостей об его уходе из Mandriva, стали перенаправлять к новому домену, http://ulteo.org, зарегистрированному годом ранее. Он содержал лишь смутные намеки на то, чем будет новый проект Дюваля, но спустя несколько месяцев появился новый сайт, www.ulteo.com, предоставляющий больше информации. На нем есть раздел новостей, форумы и сеть поддержки – вполне достаточно для поддержки выпуска нового дистрибутива. Итак, Дюваль занялся разработкой новой ОС Linux.

Мы спросили Дюваля: что было первой искрой, из которой возгорелся Ulteo? Он сказал нам, что все было как с Mandriva, но его мыслительный процесс пошел дальше: «Вдохновение было весьма схоже с умственными процессами в начале Mandrake Linux, восемь лет назад. Я просто глянул на некоторых людей, использующих ПК (в большинстве случаев под Windows) и спросил себя: что может облегчить жизнь этим пользователям? Разница в том, что с Mandrake я думал, как помочь Linux стать симпатичнее, чтобы привлечь побольше пользователей Windows. С Ulteo, я вдобавок спросил себя, нельзя ли упростить использование ОС в целом, не ограничиваясь рабочим столом.»

Linux продвинулся

Дюваль считает, что настольный Linux изменился, и это повлияло на принципы создания нового дистрибутива Linux: «Усилия, вложенные в настольный Linux, огромны. Я был очень рад видеть все эти настольные приложения, появившиеся за последние годы», – сказал он. «Я люблю *OpenOffice.org* и *Firefox*, это флагманы свободного ПО. Помню, как в 1998 я пытался убедить Corel позволить мне включить *WordPerfect 8* для Linux в Mandrake. Это было нелегко, и оно не было свободным приложением! То же было с Netscape: они только что выпустили свой код по открытой лицензии. Мы возлагали на него большие надежды, но пришлось очень долго ждать, прежде чем появилось нечто действительно полезное [*Firefox*]. Нельзя не признать, что сообщество свободного ПО немало свершило всего за несколько лет.»

Большинство дистрибутивов Linux объявляют себя легкими в использовании, и многое вправду значительно похорошело с 1998. Это не очень сильное утверждение: в конце концов, после восьмилетних трудов оно и неудивительно; но отсюда следует, что любой новой ОС Linux в 2006 не обойтись без оригинальности, выделяющей ее из толпы. Одного удобства работы уже мало. Чтобы люди выбрали новый настольный дистрибутив Linux, необходимы нововведения, которые ставят его впереди. Дюваль объясняет: «Конечно же, я многому научился по опыту Mandrake/Mandriva: и хорошему, и плохому. Ulteo предназначен быть не просто новой ОС Linux. Собственно, я надеюсь, что это будет новый способ использования ОС... хочу, чтобы Ulteo стал лучше, чем Mandriva.»

О ТОМ, ЧТО ОБЕЩАЕТ ULTEO

> «Надеюсь, что он станет новым способом применения ОС.»

Кроме произошедшего с ОС Linux, главное изменение за последние восемь лет – это наличие стабильного доступа в сеть Интернет. При выходе Mandrake это казалось наркотическими грезами, но благодаря кабельным провайдерам и ADSL большинство пользователей компьютеров в развитых странах сейчас могут позволить себе широкополосное подключение. Дюваль считает, что это облегчит сопровождение интернационального многоязычного проекта, и, как мы увидим, у него есть и другие идеи использования широкополосного доступа...

»

Манифест Ulteo

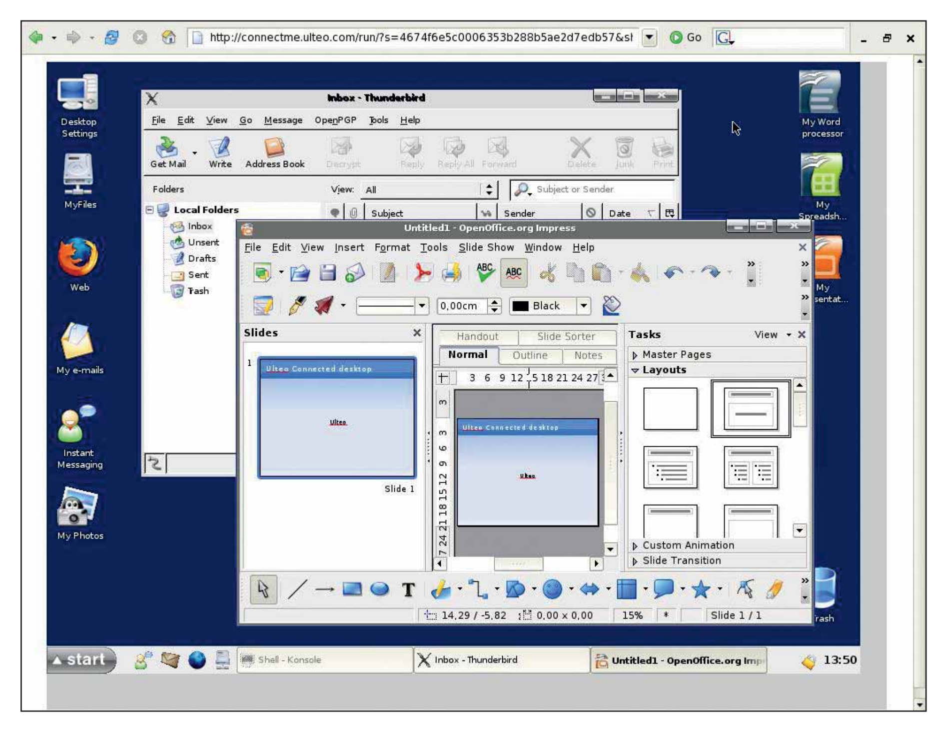


Вместо простого улучшения имеющегося дистрибутива, Ulteo метит выше – изменить способ использования компьютеров. Вот части миссии Дюваля для новой ОС:

» Освободить пользователя от традиционного системного администрирования и системных обновлений.
» Убрать необходимость установки пользователем собственного ПО.
» Предоставить силу приложений с открытым кодом пользователям Windows без принуждения пройти долгий путь установки Linux.
» С самого начала обеспечить высокий уровень безопасности.
» Обратиться к веянию времени: пользователь хочет получить доступ к своим данным и приложениям из любой точки мира.

Идеи в действии

Впечатляет? И неспроста: это наш первый эксклюзивный осмотр новой ОС.



› От Kubuntu Ulteo унаследовал лощеный интерфейс KDE и современные приложения.

Это ранний предварительный релиз Ulteo. Все выглядит в норме – используется KDE с серой палитрой и оформлением окон. На первом плане *OpenOffice.org*, а по рабочему столу разбросаны иконки. Э, да он запущен в *Firefox*! Вы не хотите перечитать последнее предложение еще раз?

Дюваль, естественно, не открывает свои карты, но вы не ошиблись, подумав, что на присланном им снимке экрана

О РАЗРЫВЕ СВЯЗЕЙ

«Я хотел загрузить Mandriva One, доступную только для членов Клуба. Я проверил свое пожизненное членство и понял, что Mandriva его удалила.»



Ulteo запущен в web-браузере. К тому же снимок показывает создание Дювалем презентации в «Ulteo Connected Desktop». Для хостинга Ulteo планируется использовать собственные сервера Дюваля, чтобы пользователи соединялись со своим настольным компьютером из любой точки мира. Вот тут и выясняется, в чем новизна дистрибутива. Системное администрирование и обновления будут выполняться на стороне сервера, как и установка приложений. [В заявлении, распространенном Дювалем после выпуска этой статьи в британской версии LXF, тем не менее, подчеркивается: «Ulteo – это обычная ОС и она может быть установлена на обычный компьютер», – прим. ред.] Уровень безопасности будет высоким, а конфиденциальность жизненно важна для успеха Ulteo. Любой сможет использовать Ulteo при помощи браузера – поистине, Ulteo не жить без широкополосного соединения – а клиенту потребуется только Java. Это означает, что пользователи Windows могут получить доступ к рабочему столу Linux через *Internet Explorer*, открыв окно в Open Source прямо с Windows-машины.

Будет ли Ulteo использовать тонкий клиент или специализированную виртуальную машину, пока неизвестно, но если вспомнить, как мало было времени у небольшой команды разработчиков Ulteo, станет ясно, что он построен на хорошо отработанной открытой технологии. Ulteo может даже объединить виртуальную машину с тонким клиентом – предоставив пользователю все преимущества реальной машины. Для серверной виртуальной машины можно использовать нечто вроде *Xen*, а для транспортировки рабочего стола Java-клиенту, запущенному в вашем web-браузере – супер-эффективный NX-протокол от NoMachine. Тогда пакет Ulteo можно продавать в духе службы «IT-департамент на час». Предприятия будут вкладывать деньги только в простые клиентские машины, подключаемые к виртуальным рабочим столам пользователей на сервере Ulteo, и платить фирме Дюваля за гарантию нужного качества обслуживания.

Еще один дистрибутив

Если вы задумали создать новый дистрибутив Linux, следует сперва решить, какой из существующих дистрибутивов взять за основу. Патрик Фолкердинг [Patrick Volkerding] строил Slackware на базе SLS (это было очень давно!); Марк Шаттлворт создал Ubuntu из Debian, а Гаэль Дюваль выбрал Ubuntu.

Вначале он планировал построить Ulteo, взяв Live-версию Mandriva – этот дистрибутив ему, очевидно, знаком. «Мои первые тесты проводились с Mandriva, по понятной причине: в этой системе Linux я разбираюсь лучше всего!» – говорит Дюваль. – «Но я также перепробовал большинство других известных дистрибутивов. LiveCD особенно интересны, это же экстракт современных Linux-технологий. Я долго колебался между Mandriva и Ubuntu, так как не знал, сколько времени займет изучение Ubuntu. Наконец, Mandriva решила за меня, выпустив финальную версию Mandriva One. Я хотел ее загрузить, но она была доступна только для членов клуба; проверил свое пожизненное членство – и понял, что Mandriva его удалила, а заодно и мой адрес электронной почты.»

Дюваль продолжает: «В ту же секунду я и принял решение: Ubuntu по духу похож на ранний Madrake Linux, и основан на Debian. Это для меня.» Точнее, окончательно за основу Ulteo Дюваль выбрал Kubuntu, поскольку KDE ему ближе (как рабочий стол первого релиза Mandrake), но KDE не единственный выбор рабочего стола. На будущее обещаны Gnome, *Xfce* и Enlightenment.

Но Ulteo обязан не просто улучшить реализацию имеющегося дистрибутива. Ему необходимо стать уникальным, предоставить то, чего нельзя найти у других. Как выразился Дюваль: «Современные дистрибутивы Linux очень хороши. Но и очень похожи. Я имею в виду, делается только объединение

компонентов, чтобы они хорошо работали и взаимодействовали. То есть интеграция. Все технологии, кроме *Xgl*, которые восхитили меня за последнее время, произошли от LiveCD. Особенно UnionFS; но виртуальные файловые системы типа Fuse также интересны и, по-моему, открыты для различных применений.»

Тайну свято сохрани

В настоящий момент, судя по результатам запросов в Google, Ulteo можно отметить как самый секретный дистрибутив Linux. Информация на сайте довольно скудная, но первое, что поражает на www.ulteo.com – это дизайн. В мире художников-любителей, профессиональный стиль оформления сайта Ulteo выделяется так же, как Вермеер среди кубистов, оставляя прекрасное впечатление: в любом рекламном агентстве именно о таком и мечтают. Но сам рабочий стол едва ли отличается от своей основы в Kubuntu. Так что же предлагает Ulteo, если он не собирается стать благообразным Ubuntu? «Наши нововведения не касаются улучшения рабочего стола», ответил нам Дюваль. «По-моему, рабочий стол Linux достиг достаточной зрелости, и у меня пока нет четких идей его улучшения или добавления чего-либо.» Амбиции Ulteo метят выше рабочего стола – изменить способ использования компьютеров!

Такая задача, очевидно, не под силу одному человеку, хотя Дюваль и считает, что проект обслуживается лучше, если соответствует видению одного человека. Ulteo зародился в беседах друзей на форумах и в списках рассылки. «Точно так же было в ранние дни Mandrake», говорит Дюваль. «Я обсуждал кое с кем технические моменты. Для Ulteo это очень хорошо, потому что когда я анонсировал проект, несколько тысяч человек немедленно проявили заинтересованность, и многие из них захотели участвовать. Ну да, я взаимодействую со многими людьми... У меня есть список того, что нужно, и я пытаюсь найти для каждой проблемы кого-то способного помочь. Вот и все. Иногда я могу кому-то заплатить за конкретное решение. Например, нелегко найти талантливого художника, желающего участвовать бесплатно.» В настоящее время еще два разработчика работают полный день, и скоро к ним присоединится третий.

Ulteo: информация



Ulteo должен выйти где-то ближе к концу этого года. Дополнительная информация – на сайте Ulteo: www.ulteo.com

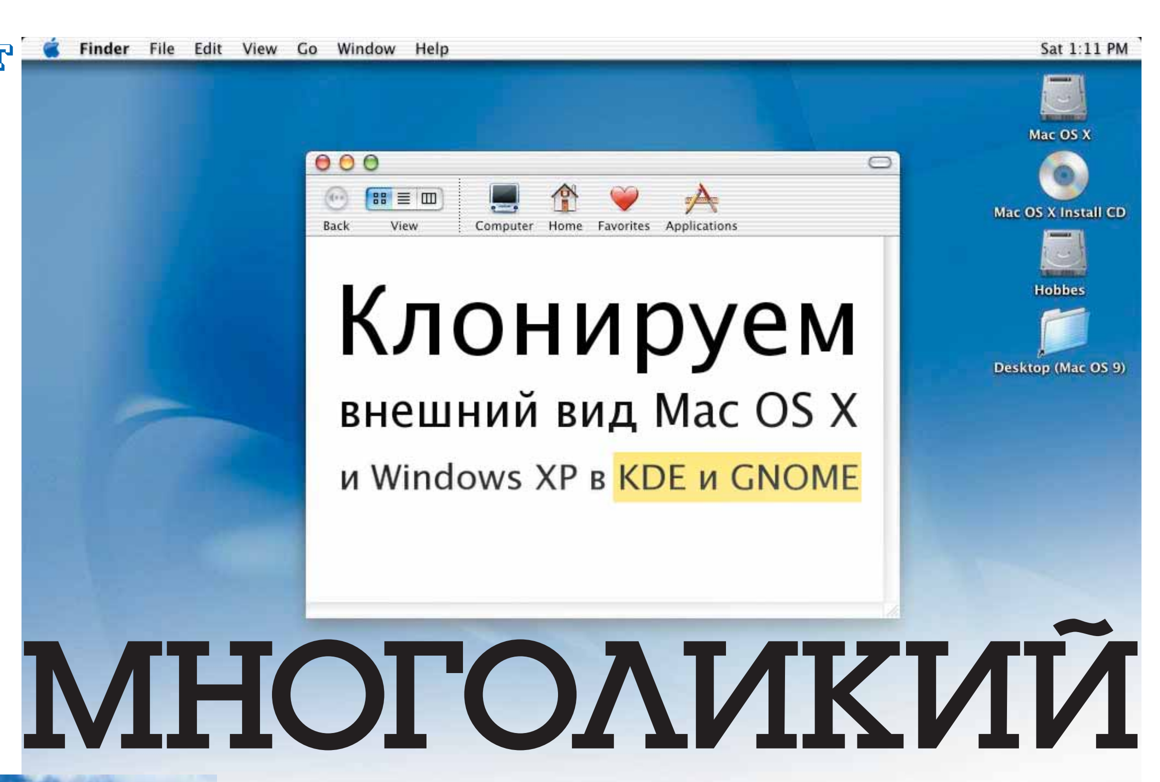
Сейчас проекту нужны разработчики Unix-PAM, эксперты в шифровании и специалисты по KDE и Gnome. Спасибо Гаэлю Дювалю за то, что он выкроил время для ответов на наши вопросы, хотя и был очень-очень занят.

Дух предпринимательства

Как ни странно, деньги этим работникам Дюваль платит из своего кармана; но он хочет поскорее привлечь инвесторов. Дюваль сказал нам: «Чем больше у нас будет разработчиков на полной ставке, тем, естественно, будет лучше. Но я умею работать и при ограниченных возможностях. Это в моей природе, и я всегда восхищался этой идеей. Инвесторы и венчурные капиталисты выглядят весьма открытыми для Linux, судя по тем людям, которые уже ко мне обращались. Но мы не будем торговать программами! У меня много идей для бизнес-моделей, подходящих для мира свободного ПО, и я пытаюсь применять их одну за другой.»

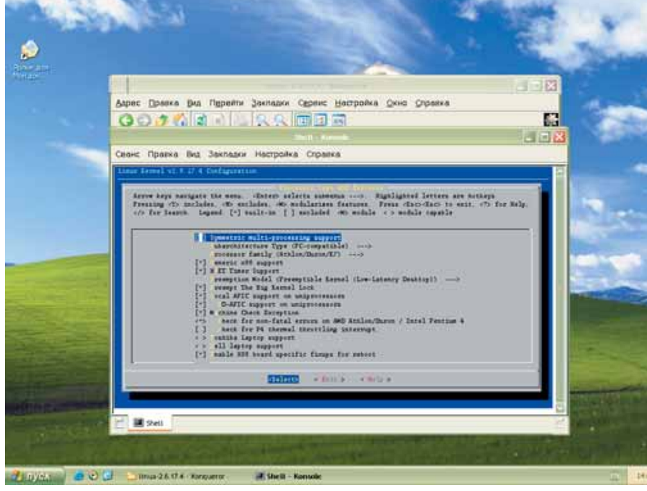
Расписание Дюваля за несколько прошедших месяцев – сериал из кошмаров. Он амбициозно пообещал выпустить новый дистрибутив Linux спустя шесть месяцев после ухода из Mandriva. Понадеемся, что к моменту чтения этих строк вы уже будете знать, изменил ли Ulteo применение компьютеров и смог ли Дюваль провернуть тот же фокус, что и с дистрибутивом Mandrake Linux почти восемь лет назад. LXF

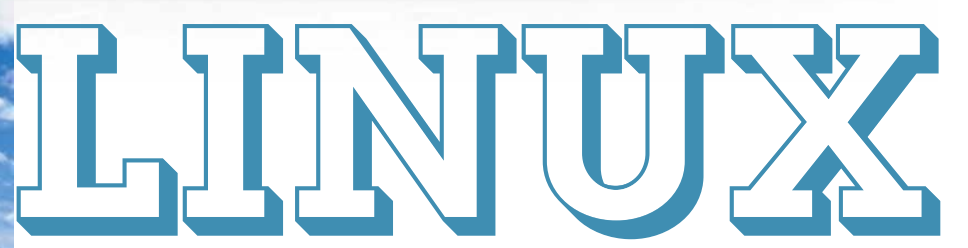




Клонируем внешний вид Mac OS X и Windows XP в KDE и GNOME

МНОГОЛИКИЙ





Вам не хватает привычного интерфейса Windows? Устали от превосходства дизайна Apple? Не расстраивайтесь – ваш Linux может с успехом копировать внешний вид других ОС, вызывая восхищение друзей и знакомых. **Александр Толстой** расскажет, что для этого требуется.

Представьте, что приятель, зашедший к вам в гости, с удивлением обнаруживает у вас установленную *Mac OS X*. Или, к примеру, сослуживцы будут по-прежнему считать, что на вашем компьютере стоит Windows XP, хотя вы незаметно подменили ее искусно замаскированным KDE. Звучит заманчиво, не так ли? Вместе с тем, вам вовсе не обязательно покупать настоящий Macintosh или продолжать мучиться с нестабильной XP. Все гораздо проще, потому что вы – пользователь Linux. В вашем распоряжении находится не только самая мощная и производительная ОС, но и бесконечные возможности по настройке и оформлению любых компонентов системы – и все это без потерь в стабильности и нарушения лицензий. Существует множество статей по оптимизации Linux, ускорению его загрузки, перекомпиляции ядра и прочим «трюкам». Нас же интересует другой вопрос: как сымитировать внешний вид и поведение другой ОС? Об этом и пойдет речь ниже.

Aqua и все-все-все...

Популярность *Mac OS X* в среде художников, аниматоров и музыкантов объясняется не только скоростью «маков», но и дизайном операционной системы OS X, которая установлена на все компьютеры Apple. OS X была выпущена в 2001 году; с тех пор и по сей день она является предметом восторженных откликов большинства пользователей. Любоваться есть на что: примененный в OS X интерфейс Aqua отличается великолепным дизайном. Сине-голубые «прозрачные» кнопки и полосы прокрутки отрисованы просто великолепно; их действительно «хочется лизнуть» – как метко выразился по этому поводу Стив Джобс шесть лет назад. Кнопки закрытия, сворачивания и разворачивания окон выполнены в виде капелек разных цветов. Они тоже являются «визитной карточкой» стиля Aqua и легко ассоциируются с «некой фруктовой компанией».

За время своего существования OS X пережила четыре обновления – с версии 10.0 (Cheetah) до 10.4 (Leopard). Изменениям подвергся и стиль Aqua, который был доработан и улучшен дизайнерами Apple.

Теперь перейдем к главному: как можно повторить это великолепие в Linux? Есть два пути – для GNOME и KDE соответственно. Для каждой из этих сред будет предложена своя методика.

Для начала выясним, что именно нам нужно имитировать, или, другими словами, из каких главных компонентов состоит Aqua. Итак, нас

интересуют: декорации окон, стиль элементов управления, иконки, указатели мыши, обои рабочего стола, настройки файлового менеджера, заставки при загрузке системы. За исключением последнего пункта, изменения настроек будут разными для GNOME и KDE.

Путь KDE

На радость всем фанатам буквы «K», разработчик под ником Thomas создал удобную тему под названием *Baghira* (**http://baghira.sf.net**), которая доступна в виде исходных текстов и пакетов RPM/DEB для основных дистрибутивов. После ее установки, в Центре управления KDE вы сможете выбрать стиль и оформление окон в стиле Aqua, а также применить нужную цветовую схему. Обратите внимание, что во вкладке Стиль имеется кнопка Настроить, открывающая обширные возможности для тонкой доводки. Вы также можете настроить поведение окон – соответствующий пункт обычно расположен в разделе Система. Зайдите на вкладку Перемещение и задайте размещение открытых окон по центру. Теперь все программы будут выровнены по середине экрана – как в *Mac OS X*.

Тема иконок не входит в состав *Baghira*, поэтому вам придется устанавливать ее отдельно. На сайте **www.kde-look.org** вы найдете несколько тем, лучшими из которых являются *OS-L* и, как ни странно, *Crystal Clear*, разработанная специально для Linspire. Последняя хоть и не полностью копирует значки OS X, но очень на нее похожа, и неплохо сочетается с *Baghira*.

Далее нужно настроить *Konqueror* таким образом, чтобы он ничем не отличался от Finder в OS X. В составе *Baghira* имеется специальное расширение для боковой панели *Konqueror*. Выберите самую верхнюю вкладку в левой части *Konqueror* (боковая панель). Затем щелкните правой кнопкой мыши по вкладке и выберите Configure > New > Add Baghira Sidebar (Настроить > Новый > Add Baghira Sidebar).

Затем нам потребуется знаменитый маковский Dock – панель внизу экрана с плавно масштабируемыми значками. Пользователи KDE могут использовать для этой цели *KSmoothdock*, *KXDocker* или *Kooldock*. Все программы можно скачать с сайта **www.kde-look.org**, там же можно найти и подробности их настройки. Нужно заметить, что эти программы (в особенности *KXDocker*) отъедают значительную часть системных ресурсов и на старых компьютерах могут сильно замедлить работу KDE. Тем, кому не так важно плавное масштабирование значков, можно посоветовать нехитрый трюк с *Kicker* – стандартной панелью KDE. Щелкните правой кнопкой мыши по пустому месту на панели и выберите пункт Настроить панель. В разделе Расположение укажите выравнивание по центру экрана, а длину где-то 70-85%, в зависимости от количества значков, которые вы собираетесь поместить на панель. В разделе Внешний вид отключите фон панели и поставьте галочку напротив пункта Включить прозрачность. Теперь осталось только зайти в Дополнительные параметры, выставить степень прозрачности (рекомендую где-то 2/3) и отключить кнопки скрытия панели. Вы также может настроить размер панели по своему вкусу, но стандартное значение Средний хорошо подходит для нашей цели. Теперь дело за малым – перетащите на панель нужные вам ярлыки и в их свойствах укажите понравившиеся вам иконки из комплектов *OS-L* или *Crystal Clear*.

Следующим шагом будет добавление верхней панели в стиле OS X. Вообще, в настройках центра управления KDE есть возможность поместить в верхнюю часть экрана строку меню текущей программы, но я рекомендую создать панель вручную. Так вы сможет добавить на нее регулятор громкости, часы, поиск файлов (например, *Kerry* – оболочку к *Beagle* для KDE) и знаменитую кнопку K. Щелкните правой кнопкой мыши по пустому месту на нижней панели и выберите пункт Добавить на панель > Панель > Панелью. Появившуюся вторую «полоску» можно перетащить куда угодно – в нашем случае на верх экрана. Верхняя панель имеет точно такие же настройки, как и ее сестричка снизу, только теперь вам не нужна прозрачность и требуется маленький размер панели. Добавим на левый край кнопку K, установив ей значок «синего яблочка». Для этого создайте на рабочем столе ярлык со следующей командой запуска: **dcop kicker kicker popupKMenu 0**. Потом ярлык можно смело перетащить на нужное место на панели. То же самое проделайте с *KMix* и *Kerry*. Вроде бы все... хотя... Мы забыли про тени! Окна в *Mac OS X* отбрасывают приятные размытые тени, которым так не хватает поклонникам Linux... Из ситуации есть несколько выходов. В KDE, начиная с версии 3.4, имеется собственный композитный менеджер – *kcompmgr*, настройки которого находятся в уже рассмотренном нами разделе поведения окон. Нас интересует вкладка Прозрачность. Включив эффекты полупрозрачности и теней, вы будете предупреждены о том, что эти функции еще не до конца отлажены и могут сбоить. Это действительно так, и вы, скорее всего, столкнетесь с неминуемым крахом X-сервера после 10-15 минут работы с тенями. Тем не менее, ничто не мешает вам использовать эти эффекты для создания потрясающих экранных снимков. Помните, однако, что для успешной работы любого композитного менеджера вам следует добавить в файл **xorg.conf** следующие строки:




```
Section "Extensions"
Option "Composite" "Enable"
EndSection
```

Не думайте, что с тенями все так плохо. Если ваш компьютер достаточно быстр и имеет процессор с частотой выше 900 МГц, вы можете использовать так называемые псевдотени. Речь идет о дополнении *Kwinshadow*, которое расширяет возможности оконного менеджера KWin. Скачать это расширение можно здесь: **http://www.kde-look.org/content/show.php?content=41064**. *Kwinshadow* работает во всех версиях KDE, вплоть до 3.5.3. После установки в разделе Декорации окон появится дополнительная вкладка Shadows, разобраться с которой не составит труда. К недостаткам *Kwinshadow* можно отнести отсутствие теней при перемещении окна, а к достоинствам – красоту и высокую стабильность.

Перейдем к виджетам. Вы, наверное, знаете, что в OS X имеется множество маленьких полезных программ, отображающих погоду, время, новости, панель воспроизведения плейера и многое другое. Все это доступно для KDE благодаря программе *SuperKaramba*. Подробно о ней уже написал Петр Семилетов в LXF 74/75. От себя добавлю, что среди тем к *SuperKaramba* вы без труда найдете аналоги маковским виджетам. Советую обратить внимание на *LiquidWeather* и *Tuxbar*.

Ну и, наконец, нам нужна заставка, отображаемая при загрузке Linux. Многие дистрибутивы – например, Mandriva, SUSE, Ubuntu и другие уже имеют свои заставки. В таком случае вам просто нужно поменять ее ему на что-то OS X-подобное....

Хуже, когда графической заставки нет, а это значит, что она скорей всего не поддерживается вашим ядром. Так обстоит дело, например со всеми версиями Fedora Core [в них используется заставка RHGB, альтернативная рассматриваемому в статье Bootsplash, – прим. ред.]. »

Степень схожести

Безусловно, у вас получится очень похоже повторить как внешний вид другой ОС, так и ее специфическое поведение. Однако, стопроцентного сходства добиться практически невозможно и подобным «моддингом» можно заниматься до бесконечности. Имейте в виду, что внимательный и опытный пользователь ПК сможет быстро заметить подвох и сообразить, что это, к примеру, Linux, а не Windows. Различия будут в любом случае. Например, скорость загрузки Windows обычно в 1,5-2 раза выше, чем у Linux. Вы можете частично решить эту проблему, используя *initng* в паре *bootsplash*, однако, все равно загрузка не станет молниеносной. Кроме этого, вам никуда не деться от фирменных особенностей GNOME и KDE, структуры *Nautilus* и *Konqueror*, которые все равно останутся сами собою. Если вас интересует не столько внешний вид Windows XP, сколько поведение, советую обратить внимание на проект XPDE (http://www.xpde.com), который изначально был призван облегчить переход пользователей Windows на Linux. Степень имитации чужой ОС здесь на высоте, поэтому не ожидайте от оболочки того же комфорта, что и от KDE. [Учтите также, что проект уже два года не подает признаков жизни, – прим. ред.].

» Выход только один – пересобрать ядро с предварительно добавленным модулем *bootsplash*. Останавливаться на этой процедуре подробно я не буду, но зато дам полезную ссылку на сайт http://www.bootsplash.org/, где можно скачать патч для ядра и ознакомиться с инструкциями.

Нам осталось установить тему курсоров для мыши и разобраться с мелочами. На сайте www.kde-look.org имеется набор под названием OS-X-Cursors, который устанавливается через центр управления (раздел Периферия > Мышь > Тема курсоров). Напоследок дам вам несколько советов:

» Заставку при запуске KDE лучше отключить совсем – в OS X ничего подобного все равно нет.

» Установите тему KDM, имитирующую приглашение OS X. Ссылка: http://www.kde-look.org/content/show.php?content=30780

» Значки на рабочем столе тоже могут отбрасывать мягкие тени. Откройте файл **~/.kde/share/config/kdesktoprc** и добавьте в раздел FMSETTINGS следующую строку:

```
ShadowParameters=0, 1, 16.0, 192.0, 2, 4, 0
```

» Используйте iTunes-подобные плейеры вроде *Amarok* и *Banshee*

» Не забывайте о темах к *Mozilla Firefox* и *Thunderbird*. Для последней программы есть отличные темы *Tiger Mail* и *Crossover* – для имитации Mail 2 и Mail 1 соответственно.

» Для того, чтобы приложения *GTK2* не выбивались из стиля, установите GTK-Qt Theme Engine (http://sourceforge.net/projects/gtk2qt/). В центре управления появится новый пункт в разделе Look-n-Feel (Оформление), и вы сможете заставить все *GTK2*-приложения использовать стиль *Baghira* (или любой другой Qt-стиль). Есть и второй способ: скачайте с сайта http://apps.gnome.org утилиту gtk2prefs. Она позволит вам задать тему *GTK2* прямо из KDE. В качестве темы задайте *Aqua Extreme Sunken* или *Mac OS X* (*подробнее о них читайте ниже*).

Наконец, все. Поздравляем с успешной имитацией OS X в KDE!

Путь GNOME

Имитация *OS X* в этой оконной среде имеет как очевидные преимущества, так и некоторые неудобства. Например, оконный менеджер *Metacity* (аналог *KWin* для GNOME) нельзя настроить на расположение окон по центру экрана, и поэтому они будут все время «липнуть» к верхнему левому углу. Вы также заметите, что цветовое оформление в GNOME привязано к теме *GTK2* и поменять какой-либо цвет можно только вручную – покопавшись в файлах настройки темы, что, согласитесь, неудобно.

Теперь о хорошем. Для GNOME существует первоклассная тема *Aqua Extreme Sunken*, которая представляет собой прямой порт стиля *Mac OS X Panther*. Скачать ее можно отсюда: http://www.users.monornet.hu/linux/GTK2/oldies/AquaExtremeSunken.tar.gz. Установка тем в GNOME порадует вас своей простотой – перетащите скачанный файл в окно выбора тем и программа автоматически предложит его установить. Тема *Aqua Extreme Sunken* выглядит гораздо красивее *Baghira* – здесь нет зазубренных краев у кнопок и прочих мелких «шероховатостей».

На сайтах www.gnome-look.org и http://art.gnome.org вас ждут обширные коллекции тем, декораций окон, обоев, курсоров и всего чего угодно. В качестве набора иконок используйте тему OS-X 3, а для декораций окон установите либо тему *Aqua Extreme Sunken*, либо *Graphite*. В последнем случае кнопки управления окном будут одинакового благородно-серого цвета и, опять же, выглядеть гораздо опрятнее, чем в *Baghira*.

Пару слов об обоях для рабочего стола. В составе *OS X 10.4* имеется подборка изображений, разделенных на три категории: стандартные, природа и абстрактные. Полный комплект вы можете скачать по этим ссылкам:

http://www.winmatrix.com/forums/index.php?showtopic=1117
http://www.winmatrix.com/forums/index.php?showtopic=4326

К слову, уже успевшие всем надоесть синие разводы относятся к разряду стандартных изображений.

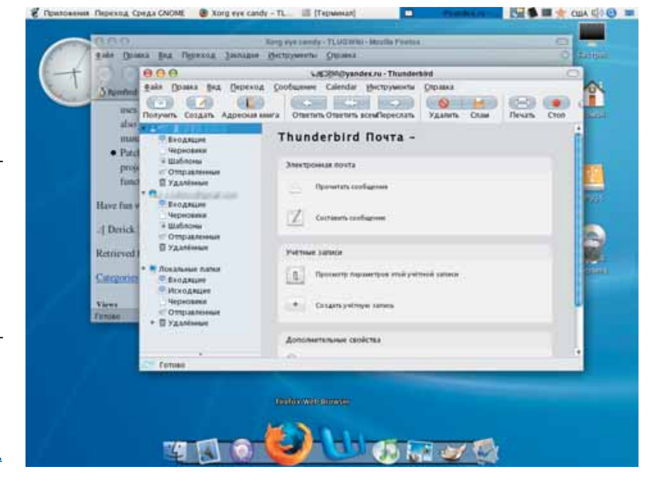
Для GNOME существует программа *gDesklets*, повторяющая по функциям *SuperKaramba* (см. LXF74/75). Виджеты здесь называются десклетами, но суть от этого не меняется – в вашем распоряжении – множество датчиков и прочих красивых безделушек, доступных на сайте

http://www.gdesklets.org/?mod=desklets/list&top

Более подробно о десклетах вы можете прочитать в уже упомянутой статье Петра Семилетова. Помимо десклетов есть еще любопытные аналоговые часы, которые работают самостоятельно, без *gDesklets* – это *MacSlow's Cairo Clock*, доступные по адресу http://macslow.thepimp.net/?page_id=23.

Что касается дока и масштабирования иконок, то здесь все более-менее аналогично ситуации в KDE. Вы можете установить десклет *Starter Bar* и наслаждаться плавными формами и невысокой нагрузкой дока на процессор. Альтернативный вариант – использовать док *Engage* из среды *Enlightenment* (http://www.enlightenment.org/Applications/Engage/) или недавнюю разработку *Akamaru* (http://people.freedesktop.org/~krh/akamaru.git/). Последняя программа не ставит своей целью повторить функциональность дока *Mac OS X*, но зато отличается потрясающими физическими эффектами, которые не оставят равнодушными даже гуру в оформлении Compiz.

Если приведенные советы показались вам недостаточными или если вы хотите как следует сбить с толку своего приятеля («что это у тебя за Винды такие?»), не поленитесь настроить на своем компьютере эмуляторы *Wine* (http://www.winehq.org) и *Dosbox* (http://dosbox.sourceforge.net/download.php?main=1). После этого, при виде одновременно запущенных *Internet Explorer*, *F-spot* и *Duke Nukem* ваш собеседник точно лопнет от зависти – или как минимум запутается окончательно.



Подражание XP

В конце концов, стиль Aqua при всех художественных достоинствах все же не так распространен и узнаваем, как жгучий сине-зеленый вид Windows XP. Не будем судить о том, насколько такое оформление красиво; ограничимся лишь тем, что и этот стиль можно без труда скопировать в Linux.

Основная тема Windows XP называется Luna; она отличается ярко-синими оттенками в декорациях окон, красной кнопкой закрытия окна и ядовито-зеленой кнопкой Пуск (Start). На сайтах www.gnome-look.org и www.kde-look.org вы найдете темы, повторяющие этот стиль (для GNOME и KDE соответственно). Иконки из стандартной поставки Windows также доступны на этих сайтах, поэтому вы без проблем справитесь с большей частью работы самостоятельно. Окна в Windows не отбрасывают теней, поэтому вам не понадобится композитный менеджер и подобные программы. Вместе с тем, важным моментом имитации Windows XP является боковая панель в Проводнике. Здесь есть смысл остановиться на KDE, поскольку файловый менеджер *Konqueror* позволяет легко модифицировать свою боковую панель. Нам нужно сделать так, чтобы на ней в интерактивном режиме отображались ссылки на актуальные в данный момент места или приложения. Чтобы добиться этого, скачайте панель *KontextBar* (http://rm.mirror.garr.it/mirrors/foxlinux/foxdesktop/updates/0.7/i386/kontextbar-0.1-5_fox.i386.rpm)

Далее нам нужно скопировать внешний вид кнопки Пуск. Для KDE существует небольшая программа под названием *KBFX*. Это специальный апплет, который можно разместить на панели Kicker через стандартный диалог Добавить на панель. У вас появится еще один аналог кнопки «K», но вызывающий приятно оформленное меню программ и разделов. Вы также заметите новый пункт в Центре управления KDE, позволяющий настроить этот апплет. При желании можно сделать так, чтобы он вызывал стандартное меню KDE, после чего кнопку «K» можно будет с чистой совестью удалить. Вы можете менять оформление кнопки *KBFX*, просто перетаскивая на нее картинки из *Konqueror*. Раздобыв снимок экрана Windows, откройте его в *GIMP* и откадрируйте до размеров кнопки. Из того же снимка вы можете взять образец заливки для фона Kicker. Для этого вырежьте из картинки небольшой фрагмент нижней панели и сохраните его в отдельном PNG-файле, после чего зайдите в свойства панели (уже знакомый вам раздел «Внешний вид») и просто укажите имя этого файла в соответствующей строке. Двигаемся дальше.

Вы вроде бы все сделали, но что-то во внешнем виде рабочего стола вас настораживает... Ах да, уберите же наконец обои с пингвином! Стандартную лужайку «Безмятежность» («Bliss»), как и весь комплект стандартных обоев XP, есть смысл переписать из установленной версии этой ОС или, если у вас ее нет, позаимствовать у друга [в последнем случае лицензионное соглашение Windows XP будет нарушено, – прим. ред.].

Вам также понадобится раздобыть стандартные шрифты Windows. По умолчанию в этой ОС используются шрифты Tahoma и MA Sans Serif, но я рекомендую установить полный комплект (http://thelinuxbox.org/?page_id=3#fonts), в который также входят Times New Roman и Arial. Впрочем, даже выставив шрифт Tahoma в настройках KDE, вы, скорее всего, заметите, что он выглядит не совсем так, как в XP. Причина в другом разрешении шрифтов в X Window и сглаживании. В LXF79 можно найти статью Владимира Попова, описывающую различные интересные манипуляции со шрифтами. Чтобы шрифты в Linux отрисовывались так же, как и в XP, вам потребуется перекомпилировать библиотеку *libfreetype* и изменить настройки разрешения шрифтов.

Теперь осталось уже совсем чуть-чуть. С сайтов оформления GNOME и KDE нужно скачать тему курсоров для мыши и изготовить *Bootsplash* с фирменным «червячком» Windows. Готовой темы в интернете мне найти не удалось, поэтому, скорее всего, вам потребуется сделать ее самостоятельно на основе уже существующей темы. Для этого вам пригодится следующая ссылка: http://linux.alhimia.ru/projects/misc/bootsplash/.

О Tahoma

Tahoma (и ее полужирный вариант – Tahoma Bold) – основные интерфейсные шрифты в операционных системах семейства Windows, которые ценятся многими за ясность и четкость отрисовки даже при малых размерах. Однако, в отличие от других шрифтов Microsoft (в том числе, Arial, Verdana и Times New Roman), Tahoma и Tahoma Bold не входят в пакет Microsoft TrueType core fonts for the Web, некогда распространяемый бесплатно (по принципу freeware) с сайта компании, а ныне живущий по адресу **corefonts.sourceforge.net**. Традиционно, эти шрифты предлагается извлекать из дистрибутива Internet Explorer, однако, поборников лицензионного ПО здесь предостерегает опасность – правом использовать данный продукт обладают только лица, имеющие лицензию на Windows.

В апреле 2006 года эта проблема (при недосмотре со стороны Microsoft) была частично решена – шрифт Tahoma был обнаружен в составе продукта Word 97 Viewer, свободного для загрузки и использования. Однако, дистрибутив Word 97 Viewer не содержит Tahoma Bold – этот шрифт по-прежнему необходимо брать из IE или обновлений к MS Office, доступных (легально) только пользователям Windows. В принципе, шрифты семейства Tahoma можно свободно приобрести в интернет-магазинах по цене около $30. Если же вы не готовы выложить такую сумму просто «за красоту» и не желаете нарушать лицензионные соглашения, попробуйте заменить Tahoma на Verdana. Разница невелика, хотя и заметна.

Если у вас возникнут сложности с установкой *bootsplash*, то возможно вам поможет статья с более детальными инструкциями: http://www.opennet.ru/base/sys/bootspash.txt.html.

И напоследок я скажу...

Прогрессивное человечество уже несколько лет не может дождаться выхода Windows Vista – очередного монстрообразного детища Microsoft. Уже известно, что в Vista будут красивые трехмерные эффекты, чем-то напоминающие XGL и Compiz, однако они будут доступны только в самых дорогих редакциях. Новый интерфейс называется Aero, и еще недавно его называли «революционным» и суперсовременным. Насколько это правдиво – покажет время, однако уже сейчас Linux предлагает все эти и многие другие эффекты для свободного пользования. Для запуска XGL и поисковой системы *Beagle* не требуется самый современный компьютер, поэтому пользователи Linux находятся в гораздо лучшем положении, чем сторонники Vista. Linux, действительно, становится все красивее и удобнее и не исключено, что в скором времени не линуксоиды будут подражать OS X и Vista, а наоборот. Кто знает, может быть, «эпоха Linux на рабочем столе» уже наступила? LXF





CINELERRA: монтаж видео в Linux

Собираетесь красиво оформить фильм о прошедшем отпуске или всерьез увлекаетесь съемками? **Петр Семилетов** расскажет о нелинейном видеоредакторе Cinelerra – свободной программе, позволяющей приобщиться к миру кино без ущерба для семейного (а, может быть, и корпоративного) бюджета.

С развитием бытовой техники видеомонтаж приблизился к людям, перестал быть делом исключительно творцов кино и работников телевидения. Если у человека есть камера и он снял множество материалов, то рано или поздно возникнет необходимость все это смонтировать, чтобы показывать зрителям в удобоваримом виде. Но такой монтаж – например, фильма о турпоходе – относительно прост, и программное обеспечение, в котором его можно выполнить, тоже относительно простое.

Однако, существует необходимость и в профессиональном ПО для монтажа. Необходимость эту испытывают как профессионалы, так и любители, снимающие собственные фильмы, в основном, игровые короткометражки и документальное кино.

Исторически сложилось так, что видеомонтаж на обычных персональных компьютерах дал сильный крен в сторону операционных систем Windows и Mac OS. Существует большое количество как ПО начального, так и профессионального уровня – *Adobe Premiere Pro*, *Vegas*, *Final Cut*, продукция от *Avid* и других компаний.

Наряду с монтажными продуктами существует также особое ПО для «композитинга», то есть составления. Этот процесс неразрывно связан с монтированием сложных кадров. Многие функции составления можно встретить в программах для монтирования. Хорошим примером специального ПО для композитинга служит *Adobe After Effects*. Состыковка двух кадров – дело монтажное, а вот чтобы поместить в пустой иллюминатор нужную вам последовательность видео, потребуется составление.

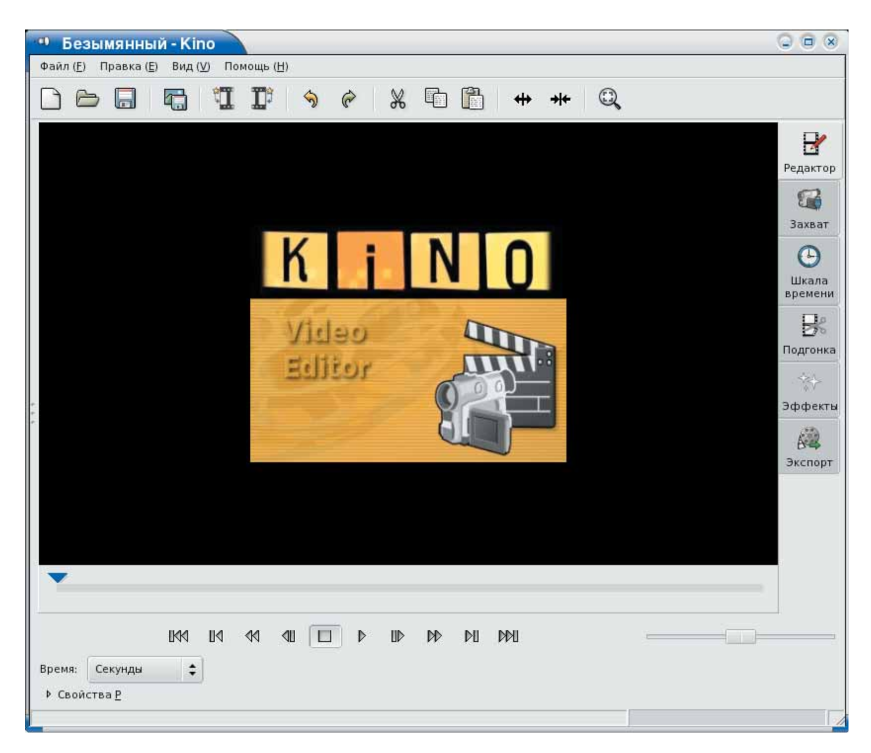
Windows и Mac – это хорошо, но как быть тем, кто в Linux? Конечно, пословица «тише едешь – дальше будешь» настраивает нас на оптимистическое видение светлого будущего, но что мы имеем сейчас? В плане видеомонтажа – не так уж много.

Пусть вас не сбивают с толку частые сведения о том, что Linux вовсю используется в процессе кинопроизводства. Это действительно так, однако, не в монтажном деле. Рендеринг трехмерной графики, обеспечение рабочего потока – да, и Голливуд, и Россия активно использует в этом деле Linux. Но когда дело доходит до монтажа – все быстренько пересаживаются на *Avid*, *Premiere* и тому подобное. Нежели Linux ничего не может им противопоставить?

Давайте разберемся. Для монтажа базового уровня, когда не требуется особых изысков, подойдет *Kino* (http://www.kinodv.org). Но у Kino есть существенный недостаток, делающий эту программу непригодной для монтирования сложных фильмов или клипов. В Kino пользователю доступна лишь одна дорожка. На этой дорожке может быть помещено только видео в формате DV (со звуком, разумеется). Дополнительных

О терминологии

Пара слов об используемой терминологии. В российском кино есть такое понятие, как «кадр». Это, грубо говоря, фрагмент видео от одной монтажной склейки до другой, либо – фрагмент от включения камеры до ее выключения. В англоязычных источниках этот же термин носит иное название – «сцена», а под кадром (frame) заграничные товарищи всегда подразумевают единичный, моментальный статичный снимок. В этой статье словом «кадр» я буду пользоваться в смысле, принятом в отечественном кинематографе, за исключением специально оговоренных случаев или когда по тексту ясно, что речь идет о «моментальных снимках» – например, в выражении «частота кадров».



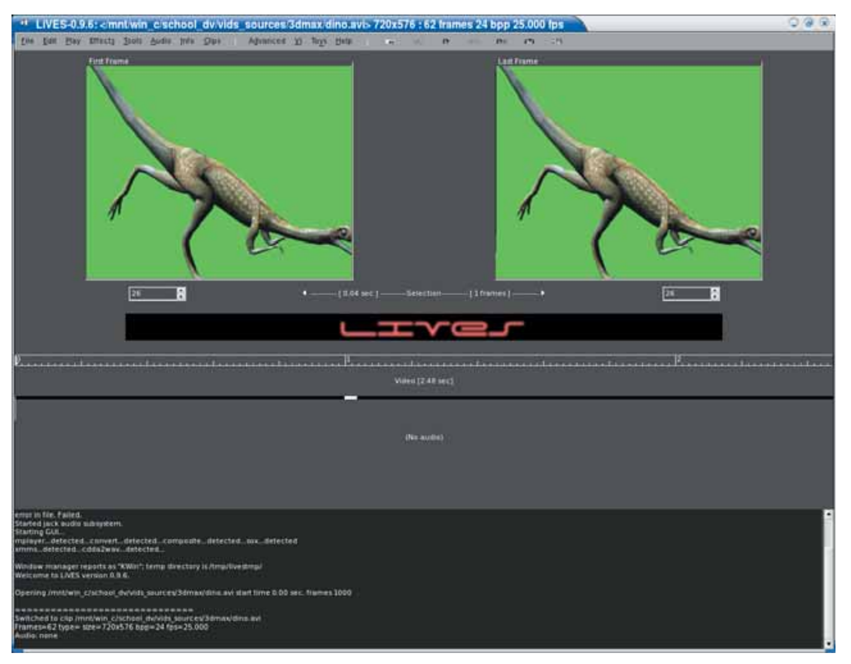
› **Окно Kino.**

дорожек ни для видео, ни для звука *Kino* не предоставляет, и это существенно сужает область его применения.

Например, существует эффект *Keying*, он же «ключевой цвет» (chroma key, color key и тому подобное). Допустим, у нас есть два кадра – один фоновый, скажем, пейзаж, а другой кадр содержит в себе объект (например, человека), снятого на однородном фоне, чаще всего ярко-синем или салатовом. Поставив эти два кадра один над другим (на отдельных дорожках) и автоматически устранив (с помощью эффекта) однородный фон второго кадра, мы как бы поместим человека на фоне пейзажа, который находится на первом кадре. Такой эффект часто используют на телевидении в прогнозах погоды или новостях.

Как видите, для профессионального монтажа нужно многофункциональное ПО. Что из этой области нам может предложить рынок Linux? Есть коммерческий *Main Concept MainActor* (www.mainconcept.com), однако он, на мой взгляд, более подходит все-таки для монтирования домашнего видео, хотя ряд функций у *MainActor* просто замечательны – это и набор эффектов, включая погодные, и встроенный MPEG2-кодек. MPEG2-кодек весьма важен для подобных программ, поскольку именно этот тип сжатия используется в DVD. Если программа не умеет выводить результаты своей работы в MPEG2, то сделать кино на DVD получится разве что выводом видео в несжатое AVI (ой как много места!) с последующим его «пережатием» в тот же MPEG2, но уже в какой-нибудь другой программе. Компания Main Concept как раз известна своими отменного качества мультимедийными кодеками, поэтому неудивительно, что в *MainActor* есть поддержка MPEG2 и захват DV-видео. Говоря о нелинейном монтаже в Linux, вспоминают обычно Lives (http://lives.sourceforge.net) и *Cinelerra*.

› **Живчик Lives.**



Я мало работал с Lives, поэтому не могу сложить о нем мнение, однако эзотерическая архитектура этой программы наводит на мысль – а нельзя ли хотя бы интерфейс приблизить к простым смертным? Другой продукт – *Cinelerra*, намного известнее и, хотя его интерфейс тоже не сахар, он более понятный.

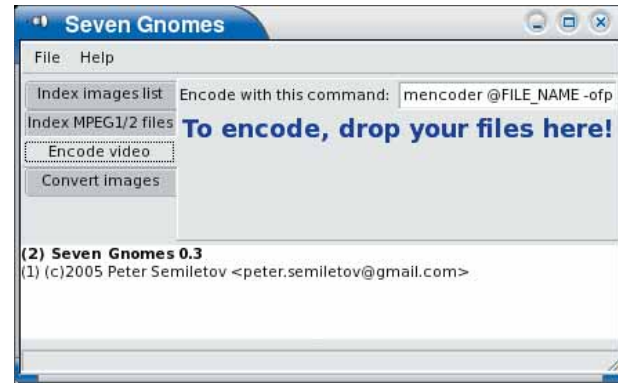
Золушка видеомонтажа

Кажется, я понимаю, почему название программы созвучно с Cinderella. За неказистым внешним видом скрыты большие возможности.

Впервые я столкнулся с *Cinelerra* летом 2005 года, когда мы с друзьями решили снимать короткометражки и видеоклипы на обычный цифровой фотоаппарат. Есть, кстати, такая мода. И снимают ведь! К счастью, позже мы стали снимать на видеокамеру, однако о начальных условия задачи я вам сообщил.

Задачи надо решать. Для этого я выполнил, как я это называю, «разработку технологии» – то есть тестовый проект, чтобы посмотреть, получится ли что-нибудь вообще. Я задумал снять пластилиновый мультфильм, который включал бы в себя покадровую (тут кадр – в смысле статичного изображения) анимацию и живое видео. Имея склонность к свободному ПО, я избрал в качестве инструмента *Cinelerra*, тогда еще линейки 1.x. Попутно я осваивал монтажное ПО и для Windows, но оказалось, что с большими массивами статичных кадров лучше всего работать именно в *Cinelerra*.

В *Cinelerra* вы можете объединять такие массивы в последовательности в виде индексных файлов (по сути, это список имен файлов, плюс некоторые параметры – частота кадров и так далее). Загружая индексный файл, вы получаете в *Cinelerra* готовый видеофрагмент, на самом деле состоящий из множества статичных элементов. Для пущего удобства я написал для некоторых присущих созданию мультфильмов операций утилиту *Seven Gnomes* (www.roxton.kiev.ua/sevengnomes.html), которая стала популярна среди сообщества *Cinelerra*.



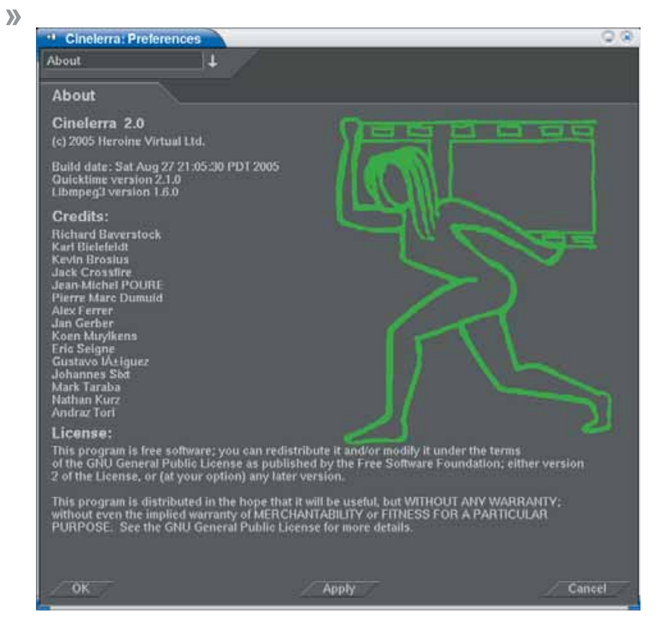
› **Утилита Seven Gnomes**

В то время *Cinelerra* распространялась в виде бинарного пакета (работающего на большинстве дистрибутивов), и в исходных текстах, которых решались собирать редкие энтузиасты. Нынче разработчики *Cinelerra* предоставляют на своем сайте (heroinewarrior.com/cinelerra.php3) только исходный код.

Существуют одновременно две версии *Cinelerra*. Одну разрабатывает коллектив Heroine Virtual Ltd. «Heroine» – это не «героин», а «героиня», поэтому на логотипе *Cinelerra* можно видеть схематичное изображение этой самой героини. »



»



› **Логотип Heroine Virtual Ltd.**

Другая версия *Cinelerra* – плод совместного труда сообщества *Cinelerra* (http://cvs.cinelerra.org). Эту версию называют *Cinelerra CV* (Community Version). Между версиями идет обмен исходным кодом. Эта статья написана исходя из опыта работы с версиями линеек 1.x и 2.x от Heroine Virtual Ltd.

Прежде чем начать разговор о том, как работать с *Cinelerra*, давайте поглядим, на что она способна и какие ресурсы ей требуются. Программы видеомонтажа всегда были одними из самых требовательных к быстродействию компьютера. Особое значение тут имеют три фактора – скорость работы процессора, объем оперативной памяти и свободное дисковое пространство. О быстром жестком диске со включенным режимом DMA я даже не упоминаю.

Разработчики *Cinelerra* советуют работать с их продуктом на 64-битных дистрибутивах – соответственно с 64-битным процессором. На нем *Cinelerra* ведет себя более стабильно. Еще разработчики рекомендуют двойное ядро. И памяти эдак гигабайта 4.

На самом же деле *Cinelerra* довольно демократична в том, что касается оборудования. Можно работать и с 512 мегабайтами памяти, и с процессором – какой есть. В отличии от, допустим, Adobe Premiere Pro, которому нужен CPU обязательно с поддержкой SSE2. Фактически, на качество результатов работы в *Cinelerra* вычислительная мощность вашего компьютера не влияет. Она влияет на работу программы в целом и на отображение эффектов в реальном времени. Но, опять же, если какие-то эффекты при просмотре у вас «тормозят», то при рендеринге фильма всё будет в порядке.

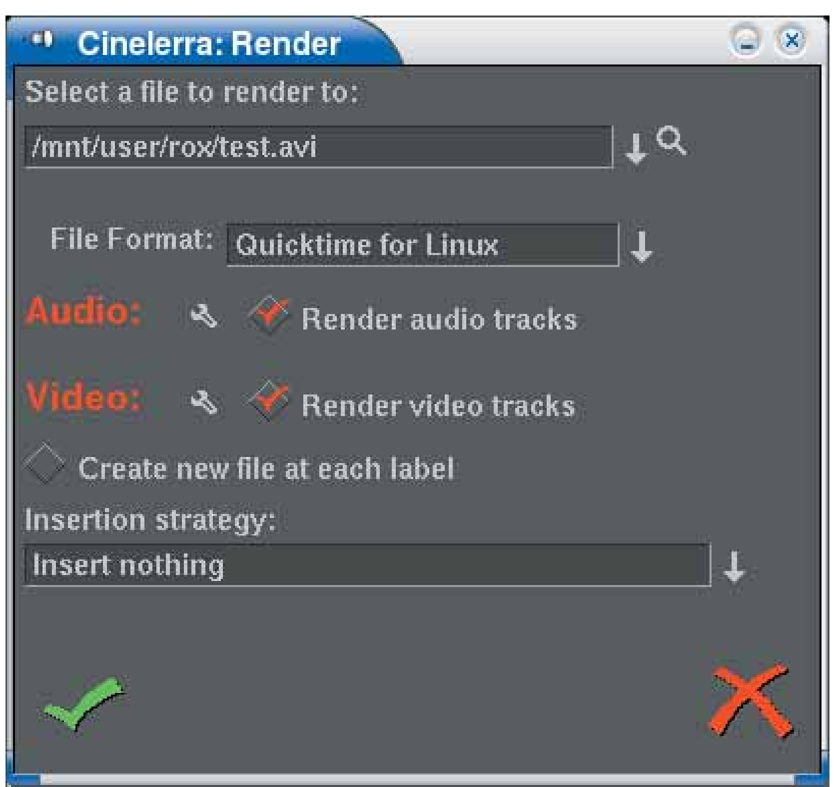
Рендеринг – это сведение всех смонтированных материалов в единое целое, в файл нужного вам формата. Какие форматы видео поддерживает *Cinelerra* для экспорта данных? Выбор широк – DV, MPEG1 (Video CD), MPEG2 (DVD), Open EXR (формат Lucasfilm), несжатое видео (с альфа-каналом), последовательность изображений (тоже с альфа-каналом), MPEG4 и другие. *Cinelerra* может экспортировать и звук, отдельно от видео.

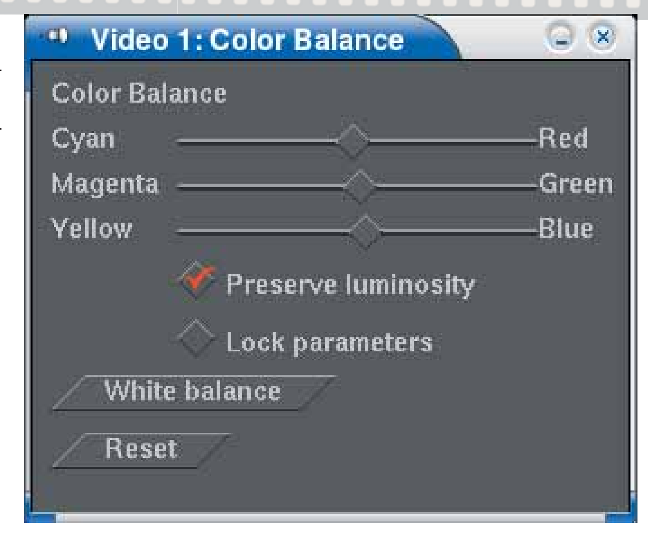
Говоря о рендеринге, нельзя обойти вниманием любопытную особенность *Cinelerra* – возможность распределенных вычислений, рендеринга по локальной сети. Что быстрее – когда фильм просчитывается на одной машине или в кластере? Вопрос риторический.

В состав *Cinelerra* входит большой набор эффектов как для видео, так и для звука. Подробно останавливаться на этом не буду – эффекты интересно пробовать, а не рассказывать о них. Отмечу лишь ряд особенностей, касающихся эффектов.

Поддерживается автоматизация параметров эффектов. То есть, вы можете изменять значения параметров на протяжении времени, и изменения будут сохранены в проекте. Управление этой автоматизацией не столь удобное, как в Adobe Premiere Pro, но все-таки оно есть и свои функции выполняет исправно.

› **Окно настроек формата вывода.**





› **Окно одного из эффектов. Все параметры можно автоматизировать.**

Далее, имеется полный комплект де-факто стандартных эффектов как для видео, как и для звука. Для звука это и ревербератор, и задержка, компрессор, эквалайзер, реверс, сдвиг питча и многое другое. Реализована также поддержка подключаемых модулей формата LADSPA. Что до видео, то здесь *Cinelerra* может порадовать нас такими эффектами, как размытие (всех сортов), chroma key, устранение шума, настройка яркости/контраста и цветовых параметров, разные декоративные штуки вроде эффекта старой пленки, живописи маслом и прочее, прочее, прочее.

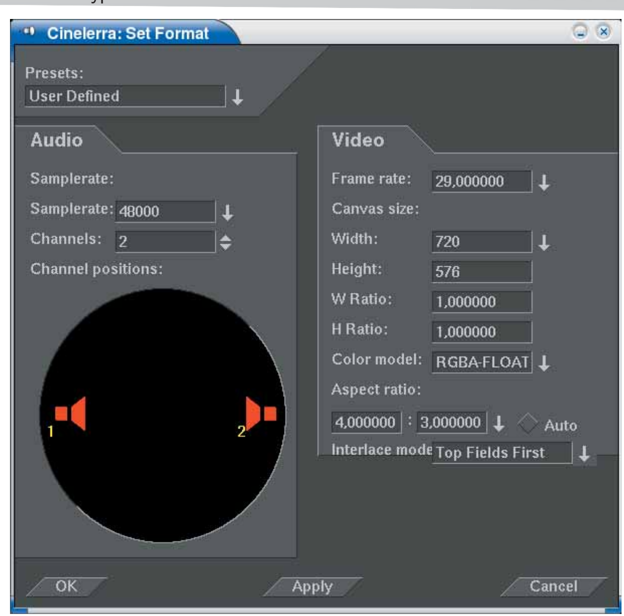
Эффектов-переходов (transitions) маловато – только стандартные. Впрочем, использование нестандартных – дело вкуса. Если вы обращали внимание, в кино кадры либо стыкуются вплотную, либо – иногда – используются переходы, такие как растворение или затемнение. В



рекламе и клипах спектр переходов, конечно, более широк. Но можно ведь делать переходы и вручную, с помощью двух дорожек и взаимодействия между ними.

Реализован движок титров, с поддержкой русского языка. Правда, в окне редактирования титры отображаются «крякозяблами», а вот на экране получаются нормально. Конечно, в Premiere Pro механизм титров более развит, но и в *Cinelerra* хорош – титры могут отбрасывать тень, двигаться в различных направлениях с разной скоростью. Есть функции выравнивания текста, установки цвета. Для простых титров этого вполне хватит, а для сложных все равно используется совсем другие программы.

Cinelerra работает с проектами в цветовых моделях RGB/A (8 битный цвет), RBG/A с плавающей точкой (32-битный цвет), а также YUV/A (8 бит). Поддерживается практически любое разрешение, соотношение сторон и частота кадров. Звук – максимум: 16 каналов, 192 КГц, 24 бита при внутренней 64-битной обработке. А это вполне профессиональный уровень.



› **Окно параметров проекта.**

Теперь, когда мы ознакомились с рабочими характеристиками *Cinelerra*, можно опробовать программу в действии, но предварительно давайте поглядим на ее интерфейс.

Интерфейс

Интерфейс *Cinelerra* построен на оригинальном наборе виджетов. Это не GTK и не Qt, хотя ничего нового для себя вы не откроете – те же кнопки, ползунки, списки. Однако интерфейс *Cinelerra* часто подвергается критике. Разработчики придерживаются своего видения построения интерфейса, поэтому на то, чтобы привыкнуть к интерфейсу героини нашей статьи, потребуется какое-то время. Я не говорю об отсутствии привычных панелей инструментов или о стандартном расположении элементов меню – это все декоративное, ни одна программа не обязана следовать документам вроде Gnome HIG (советы по построению пользовательского интерфейса, частью толковые). Но сами приемы монтирования в *Cinelerra* порой отличаются от большинства современных программ видеомонтажа. Но об этом позже.

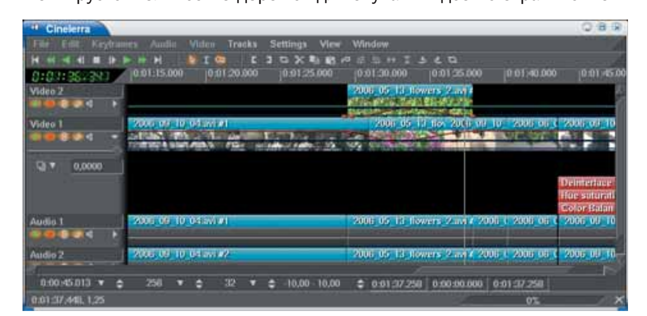


› **(Рис. 7) Основные окна Cinelerra**

У *Cinelerra* есть четыре основных окна, с которыми пользователь работает чаще всего. Это Timeline (Program), Viewer, Compositor и Resources.

1 Окно Timeline – ваш монтажный стол

Timeline (временная шкала, «монтажный стол») – окно с дорожками, на которых вы располагаете кадры и подгоняете их друг к другу, то есть монтируете. Количество дорожек для звука и видео не ограниченно.



Дорожки можно перемещать вертикально. Каждая дорожка имеет режим наложения на другую дорожку (как в *Adobe After Effects*). Помимо видео и звуковых данных, на дорожке также находятся ключевые кадры и огибающие Проектора и Камеры. Что же такое ключевые кадры?

Ключевой кадр – это, грубо говоря, набор параметров некоего эффекта в отдельной временной точке. Например, у нас есть эффект настройки контраста. И мы хотим плавно повышать контрастность от точки А до точки Б. Для этого создаются два ключевых кадра, в точке А и точке Б, и в каждой из них мы настраиваем эффект по-разному. А *Cinelerra* сделает все промежуточные изменения контрастности между этими двумя заданными точками.

О Проекторе и Камере мы поговорим ниже, когда речь зайдет об окне Compositor, пока же скажу, что огибающие Проектора и Камеры служат для точного визуального управления этими средствами. »





» Еще одна важная особенность окна Timeline – отсюда вы можете копировать части проекта или весь проект в другой проект, открытый во второй копии *Cinelerra*. Из известных мне продуктов для нелинейного монтажа на такое способен только *Vegas*.

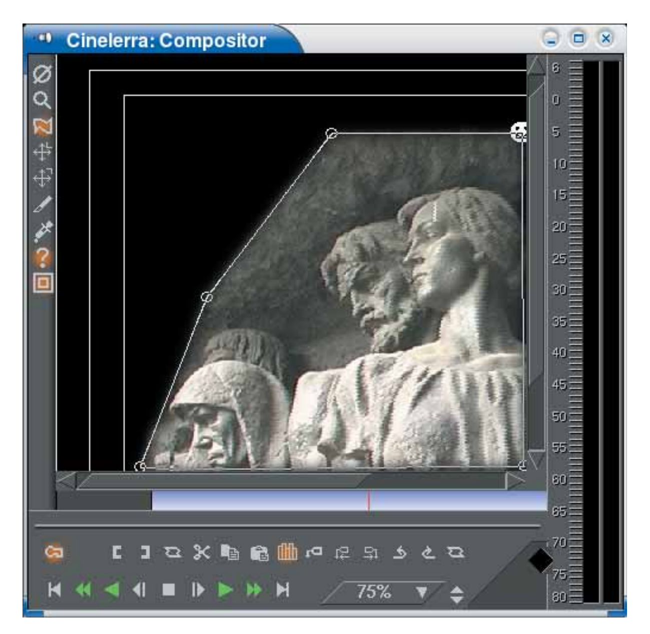
2 Окно Viewer

Окно Viewer – здесь вы просматриваете выбранный для вставки на временную шкалу материал. Кроме того, в этом окне можно выделить фрагмент и поместить его в нужное место монтируемого проекта. Во Viewer'е вы можете ставить метки и быстро перемещаться между ними, и разумеется свободно «перематывать» видео.



3 Окно Compositor

Compositor. Это окно сочетает в себе две функции. Во-первых, сюда выводится результат вашего монтажа. Иными словами, в Compositor'е отображается то, что вы монтируете в Timeline, но в том виде, в каком увидит фильм зритель. Во-вторых, в Compositor'е мы (помимо огибающих на временной шкале) управляем Камерой и Проектором. На них следует остановиться подробнее.



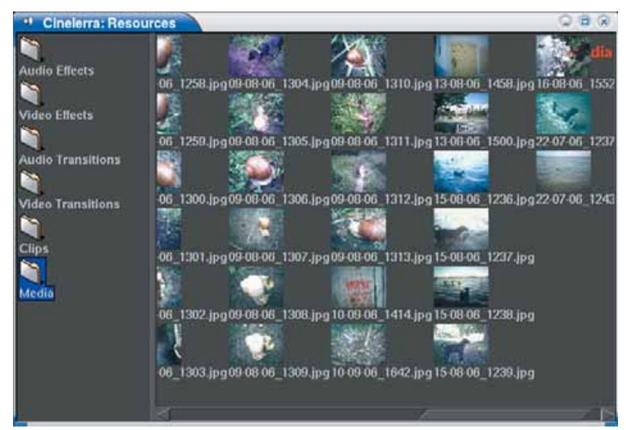
Если в *Adobe Premiere Pro*, чтобы переместить или отмасштабировать объект, мы изменяем пространственные свойства самого объекта, то в *Cinelerra* для достижения той же цели мы используем Камеру и Проектор. Настройки Проектора влияют на то, в каком месте экрана будет выведен кадр. Задействованы все три координаты – X, Y, Z (глубина). Камера же влияет на то, как расположен кадр внутри Проектора.

Допустим, у нас есть длинный статичный кадр с неким пейзажем. Весь этот пейзаж не помещается на экране и мы хотим плавно панорамировать пейзаж. В *Premiere Pro* мы бы смещали в сторону картинку с пейзажем. А в *Cinelerra* мы смещаем в сторону Камеру, как бы снимая ею нужные участки обрабатываемой картинки. Обратите внимание на крайнюю слева кнопку с изображением ключика. Она включает и выключает создание ключевых кадров. Если она включена, то все движения камеры и проектора будут «записаны» в динамике, на протяжении времени.

В том же окне – управление областью ограничения вывода (Crop) и маска. Кстати, многие инструменты окна Compositor имеют параметры, увидеть которые можно, нажав кнопку со знаком вопроса. Маска имеет два режима, один из которых, фактически, заменяет собой функцию Crop. Маска может либо ограничивать некую область изображения (а всё, что вне этой области делать прозрачным), либо наоборот, «прорезать» в картинке прозрачную дырку. Первый режим называется Multiply alpha, второй – Subtract alpha. Границы маски могут быть размытыми (параметр Feather). Масок на дорожке может быть несколько, однако все они разделяют параметры между собой. И конечно же, маска имеет такое свойство, как прозрачность (управляется ползунком Value). Подобные функции «составления» можно встретить в *Sony Vegas* и в *Adobe After Effect*, но увы – ими не может похвастаться даже *Adobe Premiere Pro 2*. Там для выполнения задач, решаемых масками, необходимо применять смекалку и множество дополнительных средств (подробнее можно посмотреть на страничке **http://www.capture.ru/adobe3/premiere2_8.html**).

4 Окно Resources

И наконец, последнее важное окно – Resources. Здесь в тематических папках доступны: эффекты видео, звуковые эффекты, переходы, клипы и материалы. Что за клипы и материалы такие? Клипы – это именованные фрагменты смонтированных кадров. Вы можете выделить область на временной шкале и сохранить ее как клип, а потом повторно использовать. Вы также можете создать клип из фрагмента в окне Viewer. Клипы доступны в папке Clips окна Resources. Здесь же есть папка Media.



Тут доступны все импортированные в проект данные – видео, звуки, картинки. Отсюда вы можете вызывать окно настроек для каждого элемента (контекстное меню, пункт Info). Это не просто информационное окно. В Info можно сменить частоту кадров, задать детали интерлейсинга и многое другое.

Вместе с тем, окно Media – самое слабое место *Cinelerra*, огромный ее минус. Дело в том, что импортированные материалы нельзя упорядочивать по виртуальным папкам. И если в проект импортировано несколько тысяч файлов, то все они будут в одной куче. Конечно же, они доступны, но как много времени придется потратить, чтобы найти нужный файл?

Как работать с Cinelerra?

По умолчанию, *Cinelerra* запускается с пустым проектом. Настройки проекта (параметры видео и звука) доступны в окне Settings > Format и могут быть изменены в любое время, даже когда в проекте уже смонтированы какие-то материалы. Импорт материалов и открытие проекта осуществляется одним и тем же пунктом меню File > Load. Файл проекта *Cinelerra* имеет расширение XML, что ясно указывает на его природу. В окне открытия файлов есть список Insertion Strategy, в котором можно выбрать, как *Cinelerra* поступит с открываемым файлом. Я советую выбрать пункт Create new resources only. С ним проекты будут загружаться как обычно, а вот видео и звуковые данные будут помещаться в виртуальную папку Media, не располагаясь при этом автоматически на временной шкале. В большинстве случаев, последнее лучше делать вручную.

Временная шкала содержит в себе дорожки. Слева на дорожке находится область управления. На ней есть кнопки, и о назначении наиболее часто используемых из них я сейчас расскажу.



› **Параметры дорожки.**

» **Play track** – включает или выключает вывод дорожки в Composer. Если эта кнопка выключена, зритель дорожку не увидит.

» **Arm track** – если выключить эту кнопку, то данные на дорожке не будут принимать участие в операциях монтирования. Если вы, например, выделяете на шкале фрагмент видео, то будут выделены данные только с тех дорожек, для которых включена кнопка Arm track.

» **Draw media** – если эта кнопка выключена, отображается только первый кадр фрагмента, если включена, то при изменении рабочего масштаба *Cinelerra* будет стараться отобразить как можно больше кадров, составляющих фрагмент.

Справа от этих кнопок находится треугольник, нажав на который, вы увидите дополнительные свойства дорожки. Это ползунок прозрачности, а под ним – способ наложения текущей дорожки на ту, что находится ниже.

Как размещать материалы на дорожке? В простейшем случае – перетаскиванием из папки Media. В той же папке можно вызвать контекстное меню и выбрать в нем команду Paste.

При вставке (Paste) объект вставляется в первую дорожку, на которой включена кнопка Arm. Однако, в какое место? В текущей позиции курсора, если термин «курсор» применим к узкой полоске, которую можно перемещать, щелкая в области «метража» на временной шкале. Однако, в случае с перетаскиванием объекта из папки Media или Clip, объект будет помещен туда, где вы его отпустите. Дальнейшее свободное перемещение объекта на дорожке реализовано довольно странно.

В прежних версиях *Cinelerra* вообще не было возможности перемещать фрагменты мышью в произвольное место на дорожке. Можно было лишь подгонять одни фрагмент к другому, да менять их местами. Чтобы выполнить монтаж, в котором фрагмент находился бы точно в нужном месте, надо было выделить определенный «кусок» дорожки и удалить его (если требовалось сдвинуть объект влево) либо добавить пустое место (чтобы сдвинуть вправо). Это выполняется пунктами меню Edit > Clear и Paste Silence, либо клавишами Delete и Shift+Space.

В community-версии *Cinelerra* был патч, позволяющий перемещать объекты свободно. На самом деле этот патч действовал точно так, как описано выше, только скрывал всю эту возню от пользователя. И вот в «официальной» версия появилась сходная возможность, однако на деле старый способ более надежен, поскольку новый позволяет переносить объекты весьма неточно.

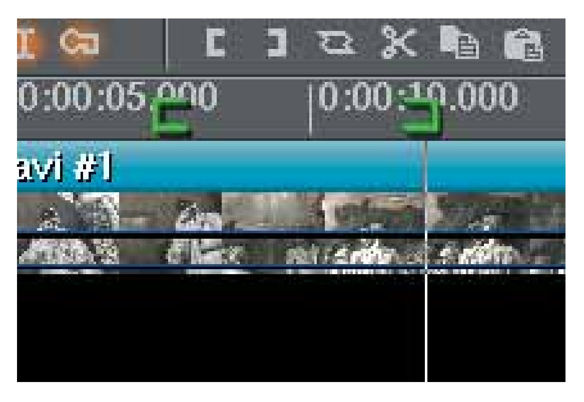
Полезные ссылки

» Руководство wiki
http://ftconsult.taxnetusa.com/twiki/bin/view/Cinelerra/CinelerraManualTOC
» Книга Secrets of Cinelerra
http://heroinewarrior.com/cinelerra/cinelerra.html

Применение эффектов.

Чтобы «навесить» эффект на расположенный на дорожке объект, надо перетащить на него эффект из папки Effect (окно Resources). Если вам нужно, чтобы один и тот же эффект длился более, чем длина объекта, вы можете заранее задать для него границы. Делается это с помощью выделения.

Выделить область можно двумя способами. Первый – это использование маркеров In и Out. Ставятся они клавишами [и], либо с помощью кнопок на панели инструментов.



› **Маркеры In и Out (скобки зеленого цвета).**

Другой способ зависит от текущего режима редактирования. В *Cinelerra* есть два таких режима. Один называется Drag and drop, а другой – Cut and paste. В последнем случае вы можете выделять данные на дорожках мышью. В режиме Drag and drop выделенной областью считается все, что заключено между маркерами In и Out. И если в режиме Drag and drop вы скопируете на дорожке данные и хотите их вставить куда-либо, то новое место вставки надо тоже отметить этими маркерами.

Здесь я описал только самые базовые приемы работы с программой – объем статьи не позволяет мне коснуться деталей. За бортом остались некоторые важные темы, например автоматизация. Но их подробно освещает документация к *Cinelerra*.

Заключение

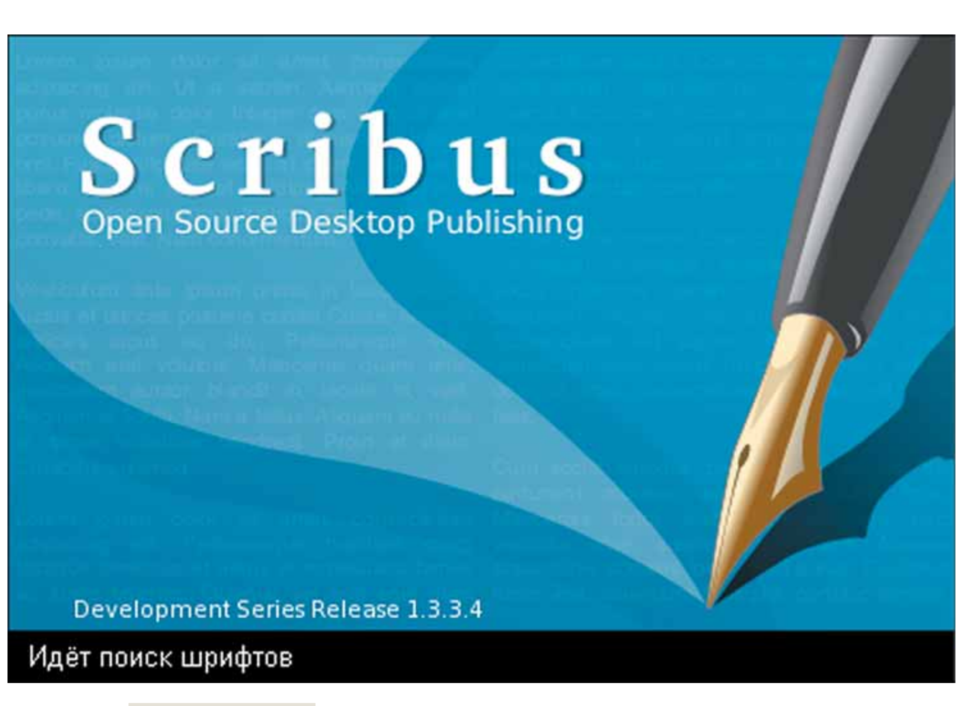
Область применения *Cinelerra* трудно ограничить. У меня нет сомнений, что *Cinelerra* технически пригодна для монтажа полнометражных фильмов, но отсутствие нормального интерфейса в области представления импортированных данных (нельзя создавать виртуальные папки, в которых удобно тематически размещать материалы) делает работу над полнометражным кино затруднительной настолько, что мало кто на это решится. Движок у *Cinelerra* очень мощный, но к нему приделан пока сравнительно слабый интерфейс пользователя, и это мешает использовать программу в больших монтажных проектах. Однако, при монтаже сравнительно небольшого количества элементов, *Cinelerra* окажется на высоте. Вы можете монтировать в ней короткометражное кино, видеоклипы и рекламные ролики. *Cinelerra* – именно профессиональное ПО и у него большое будущее. LXF

Для Вас
Хотите познакомиться с Cinelerra поближе? Пишите на letters@linuxformat.ru или www.linuxforum.ru – и, возможно, мы запустим серию учебников по видеомонтажу в Linux!



Верстаем сами

Задумали выпускать свой собственный журнал о Linux, но не знаете, с чего начать? Запаситесь качественным контентом – а направление дальнейшего движения вам укажет **Александр Супрунов**!



Скорость

» Scribus растет стремительно, обновляясь в последнее время с регулярностью, пугающей даже прожженного хакера, что уж говорить о верстальщиках! За время подготовки этой статьи пакет дозрел с версии 1.3.3.2 до идущей на диске 1.3.3.5, что и заметно в статье. А вы будете верстать с нами, наверняка, уже в следующих релизах...

Сегодня мы поговорим о верстке – точнее, настольной издательской системе *Scribus* (www.scribus.net). *Scribus* – свободное ПО, стоящее в одном ряду с такими флагманами, как *Gimp* и *Inkscape*. Его ближайший аналог – *Adobe PageMaker* – стоит $499.

Спросите себя: «Для чего мне все это может понадобиться?» Допустим, в один прекрасный день вы решите сделать рекламный буклет или выпустить небольшую книгу стихов. В этом деле незаменимым помощником окажется *Scribus*. Не захотите? Стихов не пишете? Тогда вы ошиблись уроком, начните с курса «Как писать стихи». Он подарит вам много прекрасных моментов и будет заведомо полезнее сидения перед монитором в попытках пристрелить очередного монстра на далекой космической станции.

Издательские (DTP – DeskTop Publishing) системы отличаются от текстовых процессоров из *OpenOffice.org* или *KOffice* [хотя последний обладает некоторыми функциями DTP, – прим.ред.], что, впрочем, не удивительно. Системы верстки манипулируют текстом и изображениями в виде блоков. Редактировать текст в них тоже можно, но не с таким комфортом, как в *OOo Writer*, а в *Writer* тоже можно располагать текст, но не так легко, как в специализированной программе верстки.

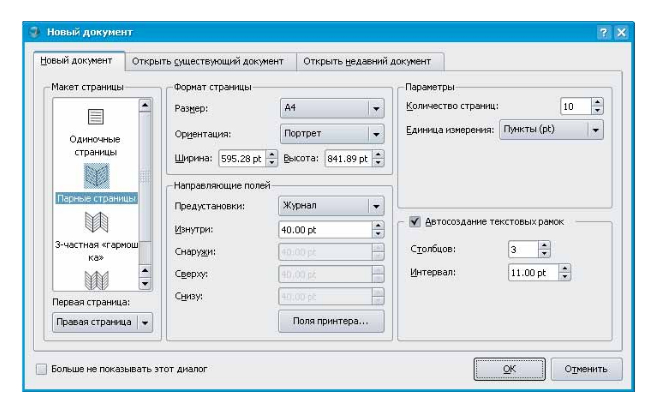
Scribus умеет сохранять результат в виде PDF (впрочем, *OpenOffice.org Writer* – тоже), в то время как многие проприетарные системы требуют для этого *Adobe Acrobat Professional* (а точнее – компонент *Distiller*) стоимостью еще $449. Если у вас от всех этих долларов уже закружилась голова, и вы решили немедленно сэкономить, следуйте за мной.

Допустим, мы создаем небольшую брошюру или журнал (кому что нравится). Для начала определимся с форматом. Здесь есть два нюанса – издание формата A4 (альбомный лист) легко напечатать на обычном бытовом принтере, который точно пылится где-то в углу во многих организациях. Если же вы решили верстать брошюру – размер более половины альбомного листа (формат A5) вряд ли вызовет нескрываемый восторг у читателей.

Итак, мы:

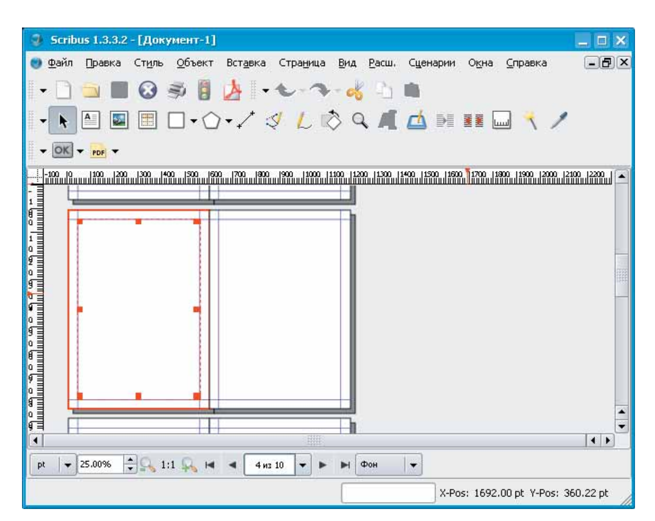
- Будем верстать журнал формата A4;
- Расположим текст в три колонки;
- Соль, сахар, перец добавим по вкусу.

Теперь самое время открыть *Scribus*. Перед вами появится рабочий стол и мастер, предлагающий выбрать параметры будущего проекта (см. **рис. 1**). В левой части мастера выберите пиктограмму «парные страницы» (для любопытных: также доступны варианты одиночные страницы / 3-частная гармошка / 4-частная гармошка). В секции «Формат страницы» укажите размер A4 и портретную ориентацию листа. Затем установите поля в секции Направляющие полей – иначе при распечатке вы с удивлением обнаружите, что текст занимает всю площадь, не оставляя места для степлера. Здесь также доступны предустановки: Ничего, Гутенберг, Журнал, Фибоначчи, Золотая середина, Девять долей.

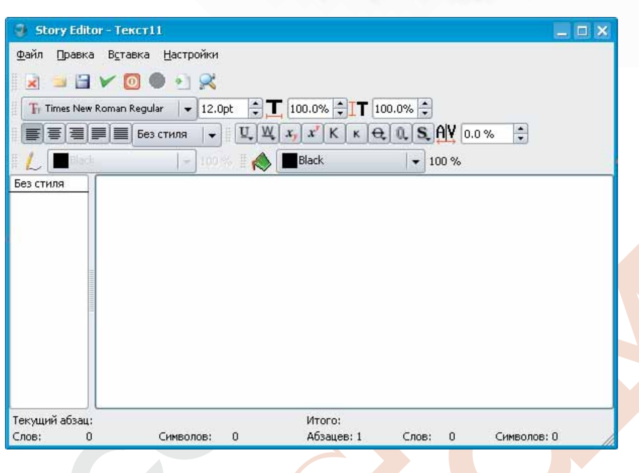


› **(Рис. 1) Создаем новый документ Scribus.**





› (Рис. 2) Рабочий стол Scribus.



› (Рис. 3) Можно приступать к набору текста.

Не будем сейчас углубляться в значение каждого пункта, отметим лишь, что Ничего – это вовсе не фамилия, а те, кто услышал знакомое слово (да, я про «журнал») – сделали правильный выбор.

Если вы уже знаете, на каком принтере будет распечатан ваш труд (пока босс вышел попить чайку или выкурить сигару), наведайтесь в раздел Поля принтера и выберите «временно ваш» принтер.

Далее нас будет интересовать секция Параметры, а именно – пункт Количество страниц.

Здесь необходимо указать предполагаемое количество страниц в журнале. Не относитесь к их подсчету чересчур серьезно – это значение можно изменить в любое время работы над проектом, а сами страницы тасовать с места на место, словно колоду карт. Для начала укажем 10 страниц.

Чуть ниже расположена неактивная секция Автосоздание текстовых рамок. Вещь достаточно полезная для автоматизации рутинных операций, поэтому отметьте ее галочкой – и вам станут доступны новые свойства. В поле Столбцов установите *3* – именно столько, как вы помните, текстовых столбцов предполагается разместить на одной странице в нашем журнале.

Управились? Теперь можно нажать кнопку OK.

Работа над проектом

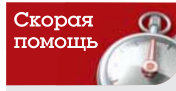
На рабочем столе *Scribus* появится стопка страниц. Перемещаться к следующей странице можно, прокручивая колесико мыши, либо перетаскивая ползунок. Обратите внимание на панель в нижней части окна. На ней в выпадающем меню 1 из 10 можно выбирать номер страницы, на которую вы желаете попасть. На этой же панели можно выбрать масштаб отображения (кроме этого, изменение масштаба происходит при удержании клавиши Alt и прокрутке колесика мыши). Я надеюсь, вы уже поняли – для выполнения многих действий в *Scribus* существует несколько путей, так что в дальнейшем мы воздержимся от перечисления всех способов решений той или иной задачи.

Очевидно, что первая страница – это обложка. Далее, скорее всего, будет слово редактора, содержание, новости, а потом дело дойдет и до статей. Вот с этих самых статей мы и начнем.

Приблизительный расчет показывает, что они начнутся страницы с четвертой, но это не столь важно – и скоро мы узнаем, почему.

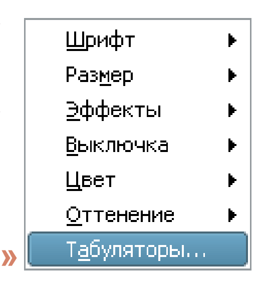
На первой странице щелкните мышью по центру листа – вы увидите красную пунктирную линию, выделяющую текстовый блок, на которой будут видны контрольные точки в виде квадратиков (см. **рис. 2**) Наведя на них курсор мыши и зажав левую кнопку, можно изменять размеры текстового блока, а для его перемещения по странице достаточно поместить курсор мыши внутрь и, нажав кнопку, тащить.

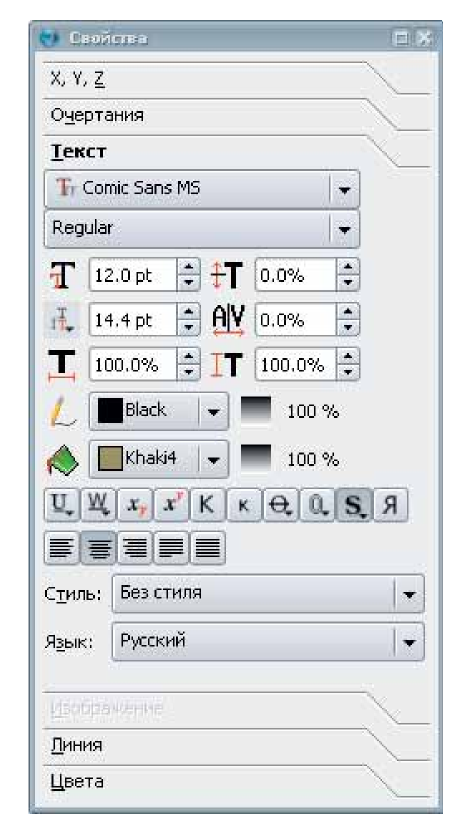
Как вы помните, мы собирались верстать наш журнал в три колонки, но на экране их пока не наблюдается. Чтобы наладить «визуальный контакт», достаточно выбрать в главном меню пункт Вид и отметить опцию Show Text Frame Columns [*Scribus* переводится на русский язык постепенно: в версии 1.3.3.4 этот пункт уже зовется «Показать границы столбцов» – прим. ред.]. »



Если ваш журнал, как и большинство молодых изданий, испытывает трудности с авторами – можете занять печатную площадь шаблонным текстом из одноименного пункта меню текстового блока. Осмысленностью он не отличается, но для экспериментов со Scribus его заведомо хватит.

› (Рис. 4) Диалог свойств объекта.





(Рис. 5) Диалог свойств объекта.

» Самого текста мы по-прежнему не видим по банальной причине – откуда же ему взяться! Вот и займемся его добавлением в текстовый блок. Щелкните правой кнопкой мыши внутри текстового блока и в развернувшемся меню выберите Изменить текст (или Получить текст, если хотите просто импортировать файл в форматах OpenDocument, HTML, *OpenOffice.org 1.x Writer*, Palm PDB или простом текстовом). Появится окно Story Editor – простенького текстового редактора (см. **рис. 3**). В принципе, можно создавать гениальную статью прямо в нем, но я надеюсь, что у вас уже все написано и ждет своего часа в файлах вроде ***.odt** или ***.txt**. Поскольку первый вариант все же распространеннее, будем считать, что материалы для вашего журнала хранятся в ODT.

Запустите *OOo Writer* и откройте нужную статью. Выделите все, все, нажав Ctrl+A, и скопируйте в буфер обмена (Ctrl+C). Вернувшись в Story Editor, вставьте текст, нажав Ctrl+V [отметим, что при этом все форматирование исходного документа будет утеряно – по крайней мере, так происходит в версии 1.3.3.2, – прим. ред.].

Теперь статью необходимо переместить из окна редактора в текстовый блок. Для этого нажмите на кнопку с изображением зеленой стрелки на листе (Обновить текстовый блок) или используйте комбинацию клавиш Ctrl+U.

Закрыв Story Editor, вы увидите на экране текст, набранный в три колонки. Но почему я не расположил его сразу же на четвертой странице? Дело в том, что, когда *Scribus* автоматически создал текстовые рамки на всех страницах, он связал их в единое целое. Добавив текст на четвертую страницу, мы бы с удивлением обнаружили его на первой. Чтобы этого не произошло, необходимо разорвать связь между текстовыми блоками на четвертой и третьей страницах. Как это сделать, будет показано ниже.

Скорее всего, текст на странице будет набран не тем шрифтом, который вы планировали использовать. Убедитесь, что текстовая рамка по-прежнему выделена и выберите пункт Стиль в главном меню (**рис. 4**).

В пункте Шрифт укажите подходящую гарнитуру – скажем, Verdana Regular. Размер – кто бы мог подумать? – позволяет указать кегль [размер шрифта в пунктах – прим. ред.], например, 9. Эффекты оставим на потом и перейдем к Выключке – она позволяет выравнивать текст в пределах колонки по левому краю, по центру, по правому краю и по ширине. Если вы не хотите, чтобы текст на странице выглядел неряшливо (это не всегда хорошо), установите выключку по ширине.

Ну вот, текст стал выглядеть более опрятно. Теперь неплохо бы расставить переносы. Это можно сделать через главное меню приложения – выберите Расш. -> Вставить переносы.

То, что получилось, выглядит весьма неплохо, но где, собственно, «шапка» – заголовок, приковывающий внимание читателей? Давайте изменим размер текстового блока – уменьшим его по вертикали, освободив место для названия статьи. Вспомните те самые контрольные точки-квадратики – цепляйте среднюю верхнюю мышью и тяните вниз. Текстовый блок станет меньше, а текст, который не уместился на этой странице, перейдет на следующую.

Пока не будем усердствовать в изысках и просто добавим в верхнюю часть страницы еще один текстовый блок, куда и впишем название. Для этого:

» Нажмите кнопку Добавить текстовый блок – она находится на панели инструментов и выглядит как лист с буквой «A» или нажмите клавишу T. Курсор мыши изменит вид на копию пиктограммы;

» Щелкните мышью по свободному месту и, не отпуская кнопки, потяните вправо-вниз, тем самым создавая текстовый блок. Щелкнув внутри блока, можно двигать его по экрану – для более точного перемещения используйте клавиатуру (клавиши управления курсором);

» Щелкните правой кнопкой мыши на блоке и выберите Изменить текст. В открывшемся редакторе Story Editor впишите заголовок статьи и нажмите Обновить текстовый блок и выйти (кнопка с зеленой галочкой).

То, что вы увидите, будет мало похоже на заголовок, но это легко исправить, используя свойства объекта. Чтобы вызвать диалог свойств, нажмите F2. Откроется окно Свойства, в котором обнаружатся вкладки:

» X,Y,Z – позволяет изменять размер объекта, поворачивать его по часовой стрелки и против, используя в качестве центра поворота либо центр рамки, либо один из ее углов. Здесь также можно создать зеркальную копию или заблокировать возможность изменения объекта.

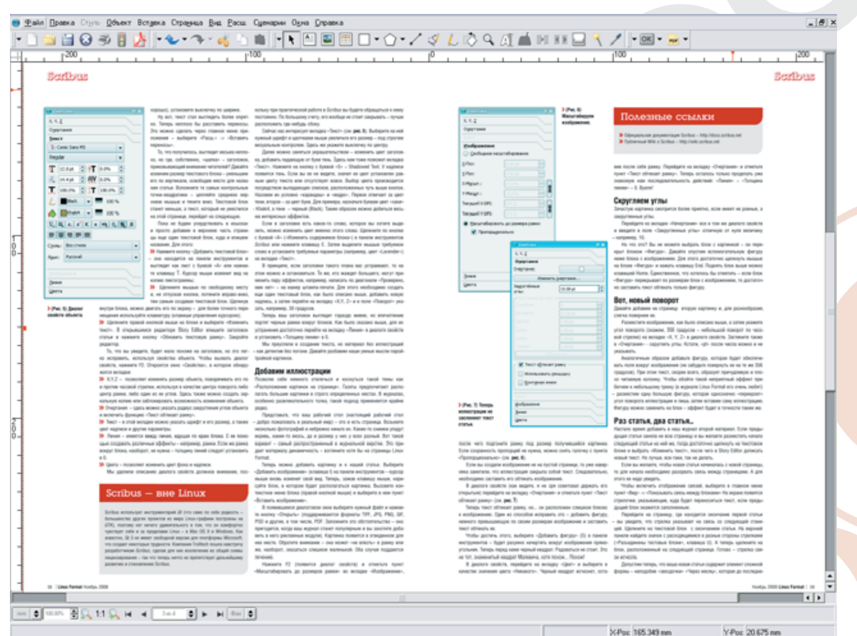
» Очертания – здесь можно указать радиус скругления углов объекта и включить функцию обтекания объекта текстом.

» Текст – в этой вкладке можно указать шрифт и его размер, а также цвет надписи и другие параметры.

» Линия – имеется ввиду линия окантовки блока. С ее помощью создаются различные эффекты – например, рамки. Если же рамка вокруг блока, наоборот, не нужна – толщину линий следует установить в *0*.

» Цвета – позволяет изменить цвет фона и надписи.

Мы уделили описанию диалога свойств должное внимание, поскольку при работе в *Scribus* вы будете обращаться к нему постоянно. По большому счету, его вообще не стоит закрывать – лучше расположить где-нибудь сбоку.



Нереальный Scribus

Вот так выглядит разворот нашего журнала в *Scribus*. Ну... если честно – так он мог бы выглядеть.

Scribus пока не до конца готов к серьезному коммерческому использованию, но он постоянно совершенствуется, и в один прекрасный день издательский мир сам не заметит, как перейдет на полный Open source!

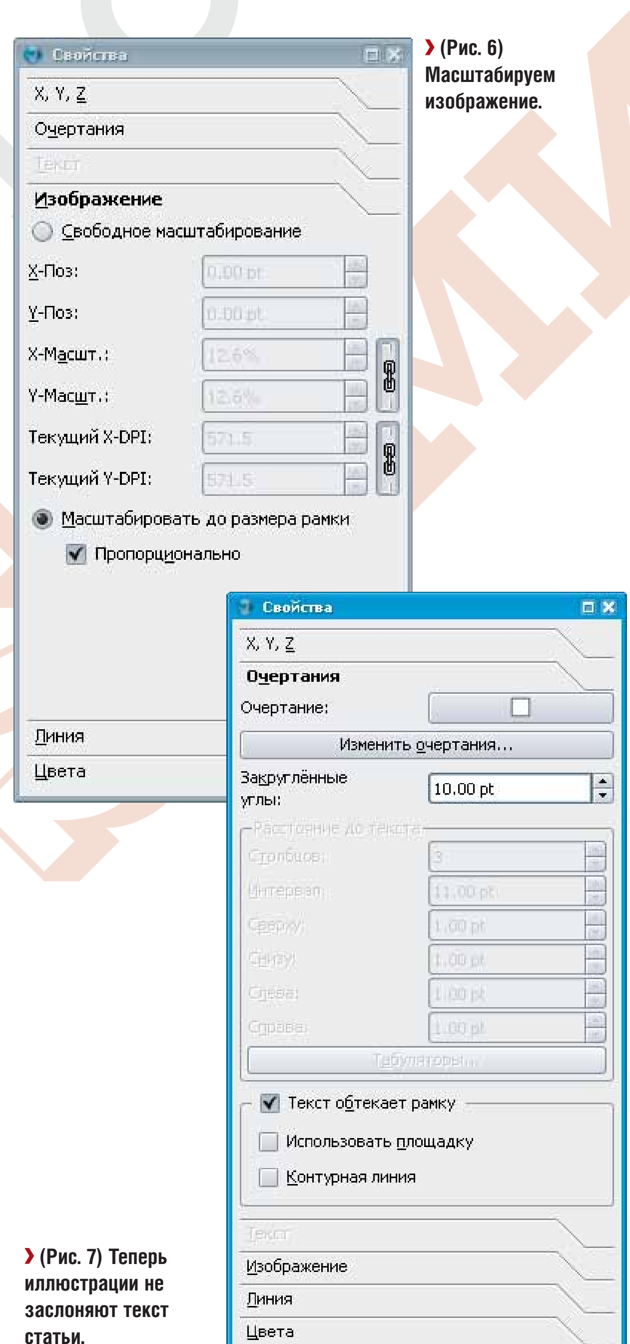
Сейчас нас интересует вкладка Текст (см. **рис. 5**). Выберите на ней нужный шрифт и щелчками мыши увеличьте его размер – под строгим визуальным контролем. Здесь же укажите выключку по центру.

Далее можно заняться украшательством – изменить цвет заголовка, добавить падающую от букв тень. Здесь нам тоже поможет вкладка Текст. Нажмите на кнопку с буквой «S» – Shadowed Text. У надписи появится тень. Если вы ее не видите, значит ее цвет установлен равным цвету текста или отсутствует вовсе. Выбор цвета производится посредством выпадающих списков, расположенных чуть выше кнопок. Назовем их условно «карандаш» и «ведро». Первое отвечает за цвет тени, второе – за цвет букв. Для примера, назначьте буквам цвет «хаки» – Khaki4, а тени – черный (Black). Таким образом можно добиться весьма интересных эффектов.

Если в заголовке есть какое-то слово, которое вы хотите выделить, можно изменить цвет именно этого слова. Щелкните по кнопке с буквой «A» (Изменить содержимое блока) в панели инструментов *Scribus* или нажмите клавишу E. Затем выделите мышью требуемое слово и установите требуемые параметры (например, цвет «Lavender») на вкладке Текст.

В принципе, если заголовки такого плана вас устраивают, то на этом можно и остановиться. Те же, кто жаждет большего, могут применить пару эффектов, например, написать по диагонали «Проверено, мин нет» – на манер штампа-печати. Для этого необходимо создать еще один текстовый блок, как было описано выше, добавить новую надпись, а затем перейти на вкладку X,Y,Z и в поле Поворот указать, например, 30 градусов.

Теперь ваш заголовок выглядит гораздо живее, но впечатление портят траурные рамки вокруг блоков. Как было сказано выше, для их устранения достаточно перейти на вкладку Линия в диалоге свойств и установить Толщину линии в *0*.



› (Рис. 6) Масштабируем изображение.

› (Рис. 7) Теперь иллюстрации не заслоняют текст статьи.

Мы преуспели в создании текста, но материал без иллюстраций – как детектив без погони. Давайте разбавим наши умные мысли парой-тройкой картинок.

Добавим иллюстрации

Позволю себе немного отвлечься и коснуться такой темы, как расположение картинок на странице. Газеты предпочитают располагать большие картинки в строго определенных местах. В журналах, особенно развлекательного толка, такой подход применяется крайне редко.

Представьте, что ваш рабочий стол (настоящий рабочий стол – добро пожаловать в реальный мир) – это и есть страница. Возьмите несколько фотографий и небрежно киньте их. Какие-то снимки упадут вкривь, какие-то вкось, да и размер у них у всех разный. Вот такой вариант – самый распространенный в журнальной верстке. Это придает материалу динамичность – взгляните хотя бы на страницы Linux Format.

Теперь можно добавить картинку и к нашей статье. Выберите Добавить изображение (клавиша I) на панели инструментов – курсор мыши вновь изменит свой вид. Далее, зажав клавишу мыши, нарисуйте блок, в котором будет располагаться картинка. Вызовите контекстное меню блока (правой кнопкой мыши) и выберите в нем пункт Вставить изображение.

В появившемся диалоговом окне выберите нужный файл и нажмите кнопку Открыть (поддерживаются форматы TIFF, JPG, PNG, GIF, PSD и другие, в том числе, PDF. Запомните это обстоятельство – оно пригодится, когда ваш журнал станет популярным и вы захотите добавить в него рекламные модули). Картинка появится в отведенном для нее месте. Обратите внимание – она может «не влезть» в рамку или же, наоборот, оказаться слишком маленькой. Оба случая поддаются лечению.

Нажмите F2 (появится диалог свойств) и отметьте пункт Масштабировать до размеров рамки во вкладке Изображение – картинка будет пропорционально растянута до заполнения ближайшего измерения блока. После этого подгоните другое измерение под размер получившейся картинки. Если сохранность пропорций не нужна, можно снять галочку с пункта Пропорционально (см. **рис. 6**) Изображение заполнит весь изначально созданный блок.

Если вы создали изображение не на пустой странице, то уже наверняка заметили, что иллюстрация закрыла собой текст. Следовательно, необходимо заставить его обтекать изображение.

В диалоге свойств (как видите, я не зря советовал держать его открытым) перейдите на вкладку Очертания и отметьте пункт Текст обтекает рамку (см. **рис. 7**)

Теперь текст обтекает рамку, но... он расположен слишком близко к изображению. Один из способов исправить это – добавить фигуру, немного превышающую по своим размерам изображение, и заставить текст обтекать ее.

Чтобы достичь этого, выберите Добавить фигуру (S) в панели инструментов – будет разумно начертить вокруг изображения прямоугольник. Теперь перед нами черный квадрат. Радоваться не стоит. Это не тот, знаменитый квадрат Малевича, хотя похож... Похож! »

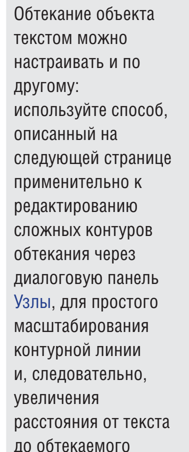
Полезные ссылки

» Официальная документация Scribus – http://docs.scribus.net
» Публичный Wiki о Scribus – http://wiki.scribus.net
» Wiki о типографике – http://en.wikipedia.org/wiki/Category:Typography
» Русская Wiki – http://ru.wikipedia.org/wiki/Категория:Типографика

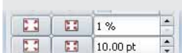




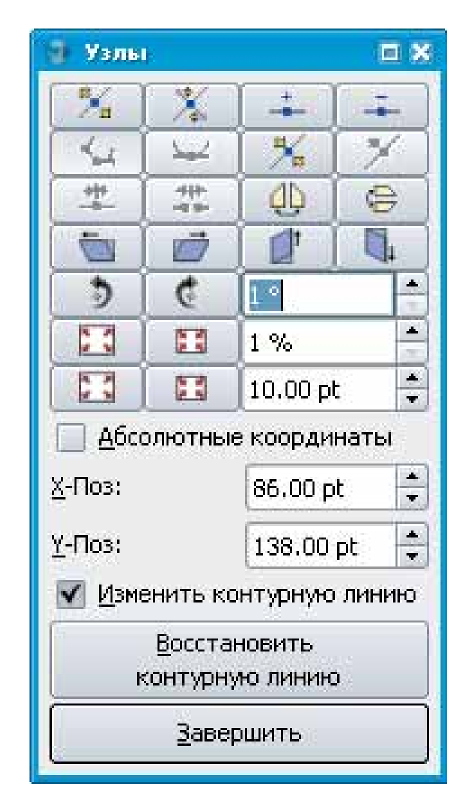
Скорая помощь



Обтекание объекта текстом можно настраивать и по другому: используйте способ, описанный на следующей странице применительно к редактированию сложных контуров обтекания через диалоговую панель Узлы, для простого масштабирования контурной линии и, следовательно, увеличения расстояния от текста до обтекаемого объекта. За масштабирование отвечают 2 нижних ряда кнопок.



Не правда ли очень просто? На самом деле, это еще и удобно. Попробуйте сами – и вам понравится!



› **(Рис. 8) Изменяем контуры объекта.**

» В диалоге свойств, перейдите на вкладку Цвет и выберите в качестве значения цвета Никакого. Черный квадрат исчезнет, оставив после себя рамку. Перейдите на вкладку Очертания и отметьте пункт Текст обтекает рамку. Теперь осталось только проделать уже знакомую нам последовательность действий: Линия – Толщина линии – *0*. Вуаля!

Скругляем углы

Зачастую картинка смотрится более приятно, если имеет не ровные, а закругленные углы.

Перейдите ко вкладке Начертания все в том же диалоге свойств и введите в поле Закругленные углы отличную от нуля величину – например, *10*.

Но что это? Вы не можете выбрать блок с картинкой – он перекрыт блоком Фигура. Давайте опустим вспомогательную фигуру ниже блока с изображением. Для этого достаточно щелкнуть мышью на блоке Фигура и нажать клавишу End. Поднять блок выше можно клавишей Home. Единственное, что хотелось бы отметить – если блок «Фигура» перекрывает по размерам блок с изображением, то достаточно заставить текст обтекать только фигуру.

Вот, новый поворот

Давайте добавим на страницу вторую картинку и, для разнообразия, слегка повернем ее.

Разместите изображение, как было описано выше, а затем укажите угол поворота (скажем, 356 градусов – небольшой поворот по часовой стрелке) на вкладке X, Y, Z в диалоге свойств. Загляните также в Очертания – скруглить углы. Кстати, «pt» после числа можно и не указывать.

Аналогичным образом добавьте фигуру, которая будет обеспечивать поля вокруг изображения (не забудьте повернуть ее на те же 356 градусов). При этом текст, скорее всего, образует причудливую и плохо читаемую колонку. Чтобы обойти такой неприятный эффект, прибегнем к небольшому трюку (в журнале Linux Format его очень любят) – разместим одну большую фигуру, которая однозначно «перекроет» угол поворота иллюстрации, и лишь затем вставим саму иллюстрацию. Фигуру можно заменить на блок – эффект будет в точности таким же.

Раз статья, два статья...

Настало время добавить в наш журнал второй материал. Если предыдущая статья заняла не всю страницу и вы желаете разместить начало следующей статьи на ней же, тогда достаточно щелкнуть на текстовом блоке и выбрать Изменить текст, после чего в Story Editor дописать новый текст. Но лучше все-таки так не делать.

Если вы желаете, чтобы новая статья начиналась с новой страницы, то для начала необходимо разорвать связь между страницами. А для этого ее надо увидеть.

Чтобы включить отображение связей, выберите в главном меню пункт Вид -> Показывать связь между блоками. На экране появятся стрелочки, указывающие, куда будет переноситься текст, если предыдущий блок окажется заполненным.

Перейдите на страницу, где находится окончание первой статьи – вы увидите, что стрелка указывает на связь со следующей страницей. Щелкните на текстовый блок с окончанием статьи. На верхней панели найдите значок с расходящимися в разные стороны стрелками (Разорвать связь текстовых блоков, клавиша U). А теперь щелкните на блок, расположенный на следующей странице. Готово – стрелка связи исчезла.

Скорая помощь

Если вы расположили все блоки так, как задумали, и не желаете случайно нарушить гармонию, любой объект можно заблокировать, т.е. сделать недоступным для изменений. Для этого выделите блок и нажмите Ctrl+L. Чтобы разблокировать объект, снова нажмите Ctrl+L.

Scribus — вне Linux

Scribus использует инструментарий *Qt* (что само по себе редкость – большинство других проектов из мира Linux-графики построены на *GTK*), поэтому нет ничего удивительного в том, что он комфортно чувствует себя и за пределами Linux – в Mac OS X и Windows. Как известно, *Qt 3* не имеет свободной версии для платформы Microsoft, что создает некоторые трудности. Компания Trolltech пошла навстречу разработчикам *Scribus*, сделав для них исключение из общей схемы лицензирования – так что теперь ничто не препятствует дальнейшему развитию и становлению Scribus.

Допустим теперь, что ваша новая статья содержит элемент сложной формы – наподобие «звездочки» «Через месяц», которая до последнего времени встречалась на страницах LXF. Как заставить текст обтекать саму звездочку, а не описанный вокруг нее квадрат?

Вставьте текст и изображение для новой статьи как было описано выше, а затем откройте диалог свойств картинки и на вкладке Очертания отметьте пункт Текст обтекает рамку. Там же отметьте галочкой пункт Контурная линия, а затем нажмите кнопку Изменить очертания. Появится панель Узлы, изображенная на **рис. 8** – отметьте на ней пункт Изменить контурную линию. Обратите внимание, что вокруг картинки появилась синяя рамка с четырьмя точками в вершинах прямоугольника. Перетаскивая эти точки-узлы, можно изменять контурную линию, которую и обтекает текст. Но четырех точек мало, поэтому в диалоге Узлы нажмите кнопку с изображением линии с точкой и знаком «+» (Добавить узлы) и щелкайте в тех местах контурной лини, где вам хотелось бы увидеть дополнительные узлы. Когда точек будет достаточное количество, нажмите кнопку Переместить узлы в том же окне и растащите узлы так, чтобы очертания контурной линии примерно соответствовали контуру фигуры.

Добавим страницу

Помните, я говорил вам, что число страниц в документе можно изменить в любой момент? Это чистая правда. Для добавления страниц следует воспользоваться пунктами Страница -> Вставить главного меню *Scribus*. В появившемся диалоге нужно указать число новых страниц (поле «Вставить») и их положение в документе.

Вставленные страницы будут лишены текстовых блоков. Не беда – всю разметку можно скопировать с уже существующих (очень желательно – чистых) страниц документа. Выделите текстовый блок щелчком мыши и нажмите Ctrl+C, а затем перейдите к неразмеченной странице и нажмите Ctrl-V.

Нам осталось преодолеть последнее препятствие – научиться создавать связанные текстовые блоки. Перейдите к одной из свежевставленных неразмеченных страниц (или удалите текстовый блок с размеченной клавишей Del) и создайте три новых текстовых блока (клавиша T). Выделите первый текстовый блок и выберите «Связать текстовые блоки» (N) в панели инструментов. Щелкните мышью на блок, в который должен «перетекать» лишний текст – появится стрелка. Проделайте ту же операцию и с третьим блоком – стрелки покажут, что теперь они образуют единое целое.

Чтобы убедиться в этом, щелкните на любом из созданных блоков и выберите Изменить текст. Скопируйте достаточно длинный текст в окно Story Editor и обновите рамку с текстом.

Поздравляю – вы успешно прошли «курс молодого бойца» *Scribus*. Он, безусловно, не всеобъемлющ, но если вы уверены, что можно сделать иначе, можно сделать лучше, чем написано на этих страницах, да и вообще все здесь перевернуто с ног на голову – я рад, что вы так думаете. Это означает, что вы не только поняли, но и проделали все то, что я вас просил, и двинулись дальше, развивая свой талант и способности. Надеюсь, мы с вами доказали, что можно комфортно верстать и в свободных программах. В этом и был смысл. Как вы думаете? LXF



Internet Control Server

О SmoothWall, IPCop и Astaro Security Firewall слышали многие. Сегодня **Михаил Журухин** расскажет вам об отечественной разработке – *Ideco Internet Control Server*

Все крупные фирмы, которые сейчас присутствуют на IT-рынке, когда-то были всего лишь start-up компаниями и начинали с малого. Обратное, к сожалению, неверно – по данным информационного агентства «Курсор», в первом полугодии 2006 г. закрылось более половины новых «start-up'ов»! Молодые компании нестабильны, подвергаются большому давлению со стороны конкурентов и… закрываются. Но некоторым удается выжить. Так, в 2002 г. небольшой коллектив единомышленников – сотрудников УГТУ-УПИ (г. Екатеринбург), рискнул выйти на рынок со своей разработкой – комплексным решением по контролю доступа в Интернет, подсчету трафика и защите локальной сети и серверов предприятия. Проект участвовал в конкурсе «Старт» Российского государственного фонда развития и предпринимательства в научно-технической сфере (фонд Бортника) и получил грант на коммерциализацию продукта.

В 2005 уже сформировавшийся продукт *Ideco Internet Control Server* (www.ideco-software.ru) получил золотую медаль на выставке в ВВЦ(ВДНХ) (г. Москва), а также был представлен и на крупнейшей в России выставке программного обеспечения Softool'2005. Сейчас *Ideco ICS* успешно развивается, и, как утверждают разработчики, уже бросил вызов Microsoft ISA Server.

В последнее время на рынке все чаще и чаще стали появляться готовые коммерческие решения на базе ОС Linux. Это и брандмауэры, и маршрутизаторы, почтовые шлюзы и прочие системы класса «поставил и забыл». Обычно такие системы стоят где-нибудь в углу серверной комнаты и работают круглосуточно, передавая пакеты из локальной сети в Интернет и обратно.

Ideco ICS относится к классу подобных систем и является Интернет-шлюзом, который предоставляет доступ в сеть всем пользователям. Сегодня организация, столкнувшаяся с проблемой учета трафика и контроля доступа при использовании Интернет-канала, может пойти несколькими путями:

» Установить аппаратный маршрутизатор или Firewall + компьютер с программой обработки статистики и управления.

Компоненты Ideco ICS

Linux 2.4.24 i686 SMP.	firebird 1.0.3.972-0.64IO.
poptop-1.2.1.	iptables 1.2.9a-2.
pppd 2.4.2.	dnrd 2.17.
thttpd 2.23.	dhcpd 3.0pl1-23.
postfix 2.1.5.	php 4.3.8.
teapop 0.3.8.	vsftpd 1.1.3.
openssh-3.5.	ntpd 4.1.1c.

Если есть несколько вариантов, используются наиболее компактные и простые компоненты, т.к. огромный функционал их собратьев-гигантов просто не нужен. Хорошим примером может быть thttpd (77 Кб) или dnrd (50 Кб).

Почему именно Firebird?

«Из всех открытых решений, в качестве базы данных был выбран *Firebird 1.0* (реализация *Interbase*). Эта база данных версионного типа, и вся информация о транзакциях хранится прямо в БД, а не в промежуточном журнале, поэтому в случае аппаратного отказа, восстановление БД происходит очень быстро и в подавляющем большинстве случаев без потерь. *Firebird*, по сравнению с *MySQL*, более стабилен и изначально поддерживает хранимые процедуры, что обеспечивает целостность данных. По функционалу *Firebird* ближе к таким БД, как *PostgreSQL*, но при этом не требует сопровождения. Вся информация хранится в одном файле, что уменьшает вероятность критических сбоев.

Вообще, *Firebird/Interbase* оптимально подходит для встраиваемых систем: например, он применяется в США в авиастроении и даже в бортовой системе американских танков».

Марк Коренберг, ведущий разработчик Ideco Software

» Использовать собственные разработки на базе Linux/FreeBSD.
» Использовать программы и службы под Windows.
» Развернуть комплексное решение, реализующее необходимые функции.

С первыми тремя пунктами все ясно: все системные администраторы, так или иначе, имели с ними дело. Последний пункт – самый заманчивый. Комплексных решений, тем более на базе ОС Linux, не слишком много, и *Ideco ICS* является интересным представителем этой категории продуктов. В числе его возможностей: фильтрация трафика, блокирование рекламы, интеллектуальная приоритизация трафика (QoS), удаленное подключение сотрудников и подразделений, почтовый сервер с антивирусом, корпоративный web-сервер, DHCP, шейпер трафика, FTP-сервер.

При этом все компоненты *Ideco ICS* легко настроить. Интерфейс продукта можно назвать дружественно-интуитивным, поэтому *Ideco ICS* может настроить и человек, не знакомый с Linux.

Установка и управление

Установка происходит на чистый компьютер, т.к. *Ideco* не соседствует с другими ОС. Инсталляция происходит в автоматическом режиме и не отнимает много времени. В результате получается уже настроенная система, остается только произвести «доводку»: ввести пароль (по умолчанию – «servicemode») и получить доступ к настройкам. В разделе «Конфигурирование сервера» вводятся IP-адрес, маска подсети, имя сервера, параметры VPN и DNS.

Управление сервером осуществляется с любой рабочей станции Windows с помощью *Ideco ICS* Manager или через web-интерфейс. Здесь создаются тарифные планы, происходит управление пулами IP-адресов, добавление и удаление пользователей. Теперь, когда система работает, самое время узнать, как и почему.

Принцип работы

Сервер имеет два сетевых интерфейса: один интерфейс подключен к Интернету, а второй – к локальной сети предприятия. Принцип работы хорошо известен: сервер подключается к Интернету и имеет реальный IP-адрес. Все остальные компьютеры получают доступ к сети через шлюз по технологии NAT (Network Address Translation) и имеют «фиктивные» адреса, чаще всего, вида 192.168.x.y. В случае необходимости, сервера, расположенные за периметром брандмауэра, могут получить через VPN реальный IP-адрес. Поддерживается и публикация серверов с помощью отображения портов DNAT.

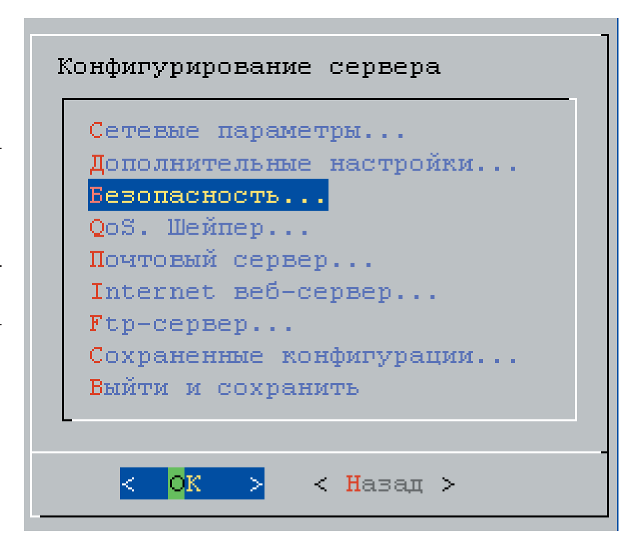
Пользователи подключаются к шлюзу с помощью технологии VPN, что позволяет ввести надежную схему аутентификации. Учет трафика также ведется по пользователям, а не по компьютерам (IP-адресам).

Иногда у пользователей локальной сети, находящихся в другом городе или просто вне здания организации, возникает потребность обратиться к ресурсам родной локальной сети. Любому пользователю *Ideco ICS* можно разрешить доступ из Интернета к локальной сети; при этом VPN-канал, разумеется, шифруется.

Внутреннее устройство

В основе Ideco Internet Control Server лежит Red Hat Linux 9.0, однако, за время своего существования продукт был практически полностью переделан. Система включает в себя такие компоненты, как ядро 2.4.24 i686 SMP, poptop-1.2.1 (открытая реализация протокола PPTP, используется для создания VPN), база данных Firebird (открытый аналог Interbase) и другие. В системе присутствуют и модули собственной разработки: основной демон управления в реальном времени, который управляет подключением-отключением пользователей, правилами iptables, подсчетом баланса и синхронизацией с базой данных, а также сборщик и агрегатор статистики.

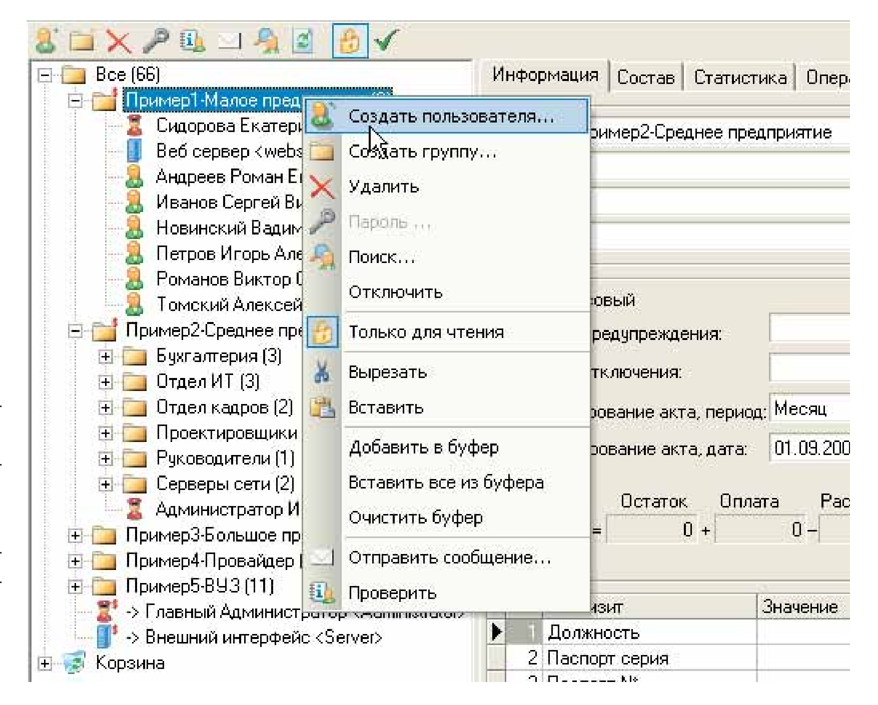
В процессе установки системы жесткий диск будет разбит на четыре раздела. На одном из них стоит сервер, с него же загружается система. Этот раздел не изменяем. На втором разделе содержатся данные: здесь находятся файл конфигурации и база данных пользователей со статистикой. Однако, с этого раздела запрещено исполнять файлы.



Это сделано по соображениям безопасности. Вообще же безопасность в этой системе стоит на первом месте.

Посудите сами. В системе нет пользователя root, и даже администратор работает не от root и только в **chroot**-окружении. Многие открытые компоненты, в том числе ядро Linux, подверглись переработке. Ядро Linux было пересобрано, и на него наложены security-патчи (как стороннего, так и собственного производства), которые предоставляют дополнительные возможности. Эти возможности активируются через ветку **/proc/kmask**. Перечислим наиболее интересные из них. Опции включаются записью единицы в соответствующий файл. Отменить опцию невозможно – даже пользователю root.

» **Noexec** – после установки этого параметра, все монтируемые разделы будут подключаться с флагом noexec.
» **Nosuid** – аналогично noexec, но запрещается создавать файлы с флагом suid.
» **Nodev** – запрет задания файлов устройств на разделе и запрет монтирования devfs.
» **Nounchroot** – запрет пользователю root делать **chroot** на корневой каталог, то есть запрет выхода из **chroot**.
» **Noproc** – запрет монтирования procfs.
» **Noaddmount** – запрет монтирования новых разделов.
» **Capmask** – дополнительная маска. После ее установки нельзя выйти за рамки возможностей, разрешенных этой глобальной capmask. »





» Используется монолитное ядро, собранное без поддержки модулей – во избежание внедрения руткитов.

Особое внимание также уделяется защите системы и локальной сети от атак злоумышленников. По умолчанию, все правила системного брандмауэра настроены на максимальную защиту и ни один порт сервера недоступен из Интернета. Используется уже стандартная на сегодня политика типа «deny, allow», то есть все, что не разрешено явно, запрещается. При этом через пользовательский межсетевой экран нельзя создать правила, уменьшающие безопасность сервера *Ideco ICS* – только правила-ограничения для пользователей.

Имеется функция фильтрации содержимого, позволяющая запретить загрузку из Сети определенных типов файлов (например, MP3, AVI и др.) и рекламы. В брандмауэре можно включить качественный шейпер с очередями, это позволяет ограничить ширину канала для конкретного пользователя или протокола. Такими возможностями сейчас никого не удивишь, во всех современных шлюзах они должны быть.

Права и возможности служб (capabilities) изменены и максимально ограничены, что уменьшает вероятность внедрения эксплойтов, т.к. с установленными значениями capabilities выполнить запрещенные действия становится весьма затруднительно.

Используя стандартные capabilities ядра Linux, можно гибко ограничивать возможности процессов, что особенно важно для приложений, работающих от имени root. Установка производится либо для всех создаваемых процессов командой lcap, либо для каждого процесса отдельно командой setpcaps. Администраторы Linux редко пользуются capabilities и, на взгляд разработчиков *Ideco ICS* – напрасно. Это стандартная возможность ядра, которая значительно повышает защищенность системы в целом.

Для улучшения безопасности системы используется технология chroot. Chroot – это всего-навсего каталог, содержащий миниатюрную копию файловой системы Linux. Поддерево chroot содержит только те файлы, которые необходимы для работы нашей службы. Служба, запущенная в chroot, не влияет на корневую файловую систему, и другие программы, которые выполняются не в chroot-окружении, не могут повлиять на работу службы. В *Ideco ICS* используется специальный усиленный chroot – root не может выйти за его пределы, поэтому, если будет найдена уязвимость в одной из служб, это не приведет к нарушению безопасности всей системы.

При старте службы ей устанавливается приоритет. Службы, отвечающие непосредственно за передачу пакетов, имеют наивысший приоритет, а, например, агрегатор статистики – наименьший.

Так как root в системе отсутствует, некоторые административные операции производятся через модуль обработки системных команд – execd. Все подобные задачи: отправка уведомлений, перезагрузка сервера, создание резервных копий и т.п., становятся в очередь сообщений POSIX. Execd читает данную очередь и выполняет задачи последовательно. Довольно необычный подход, в Linux традиционно используется *sudo*; однако здесь администратор находится в chroot, и *sudo* не поможет.

Статистика в *Ideco ICS* собирается демоном собственной разработки. Демон работает с приоритетом реального времени через интерфейс *ulog* и сохраняет статистику в файлы, которые затем в два этапа обрабатываются агрегатором.

В сервер встроена возможность периодического резервного копирования БД и конфигурационного файла на CD-диск. Стоит отметить, что вся конфигурация сервера находится в одном файле, что упрощает восстановление при аппаратных сбоях или переносе системы на другой компьютер.

Для повышения надежности и отказоустойчивости системы используется *Watchdog*.

Watchdog работает как виртуальный администратор: следит за работой системы, останавливает и перезапускает «подвисшие» процессы, при необходимости осуществляет «мягкий» перезапуск системы и посылает сообщения администратору.

Почему Ideco Software не использует новейшие компоненты?

«Главным принципом *Ideco ICS* является надежность. Не скорость, эффективность, малый размер и современность, а надежность и стабильность. Можно провести аналогию с «допотопными» космическими ракетоносителями, разработанными в 60-х годах – на сегодня это самый надежный способ выхода в космос, проверенный годами.

Аналогично дела обстоят и с программным обеспечением. По нашему мнению, главным фактором, определяющим надежность программного компонента, является долгая активная работа на множестве серверов по всему миру. Как только обнаруживается сбой, сообщение об этом передается автору, и он исправляет ошибку. Таким образом, выпускаются небольшие «патчи», исправляющие данную конкретную проблему. В таких случаях изменения исходного кода минимальны и не касаются всей программы в целом, а значит, не порождают новых ошибок. Кроме этого, мы выбираем только те разработки, исправлений по безопасности в которых уже не было более двух лет.

Самые современные версии программ обладают новой функциональностью и иногда даже базируются на новой архитектуре или вообще созданы с нуля. Так или иначе, в новых исходных текстах содержатся новые ошибки. К тому же для наших целей не нужна новая функциональность – вполне хватает и стандартной. Таким образом, нет ни одной причины использовать новейшие версии программ во встраиваемых и необслуживаемых системах.

Конечно, когда нам понадобится новый функционал, придется использовать новую версию, но и при этом наши разработчики сначала тщательно просмотрят изменения между версиями и только затем начнут ее тестирование и использование».

Марк Коренберг, ведущий разработчик Ideco Software

Коммерческий проект и сообщество Open Source – точка взаимодействия

«Чтобы исправить ошибки, зачастую принималось решение не изменять исходный код, а, зная об этих недоработках, исправлять поведение системы так, чтобы она не страдала от проблем открытых компонентов. Фатальных ошибок в программах мы пока не находили – только неоптимальные и неэффективные алгоритмы. Сделанные нами изменения касаются как ядра (*iptables* – модули psd, connlimit, string), так и приложений (*pppd*, *postfix*, *thttpd*). Пришлось писать новые модули для доступа к БД *Interbase* для *postfix* и *teapop*, так как официальные коды поддерживают подключение к этой БД. В *vsftpd* добавлена возможность контроля свободного места на диске, а в *postfix*, помимо прочего – возможность контроля за размером каталога с письмами.

На практике оказалось, что разработчики Open Source, как правило, реагируют на наши сообщения очень медленно и неохотно. Все дело в том, что мы имеем дело со очень специальными случаями, которые проявляются при больших нагрузках. В обычных условиях то, на что мы наткнулись, не проявляется, поэтому такие ситуации сложно выявить и провести повторные эксперименты.

Тем не менее, многие сообщения об ошибках были подтверждены и исправлены разработчиками.

Все изменения и патчи мы высылаем по запросу на e-mail в течение двух дней, в соответствии с лицензией GPL».

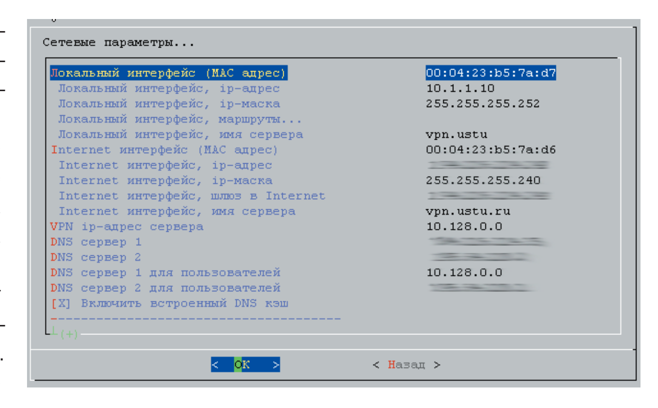
Марк Коренберг, ведущий разработчик Ideco Software

Для отслеживания состояния служб *Watchdog* использует PID-файлы в каталоге **/var/run**. Если PID-файл службы есть, а процесс не найден, то служба перезапускается. Для всех процессов проверяется объем используемой памяти и потребление процессорного времени. Если процесс превысил установленные лимиты, то он уничтожается. Также *Watchdog* проверяет свободное место на диске и другие критические параметры, и в случае необходимости предупреждает администратора или, в крайнем случае, переводит систему в runlevel 2 – своего рода SAFEMODE.

При загрузке система проводит самотестирование: проверяет параметры оборудования, состояние файловой системы и базы данных, а также сверяет контрольные суммы всех исполняемых файлов. Раздел с исполняемыми файлами монтируется в режим ro, а после загрузки всем исполняемым файлам выставляется атрибут Immutable и общий системный флаг Immutable. После этого даже root не может изменить системные файлы. Вельма разумный подход; удивительно, почему популярные дистрибутивы Linux его не используют.

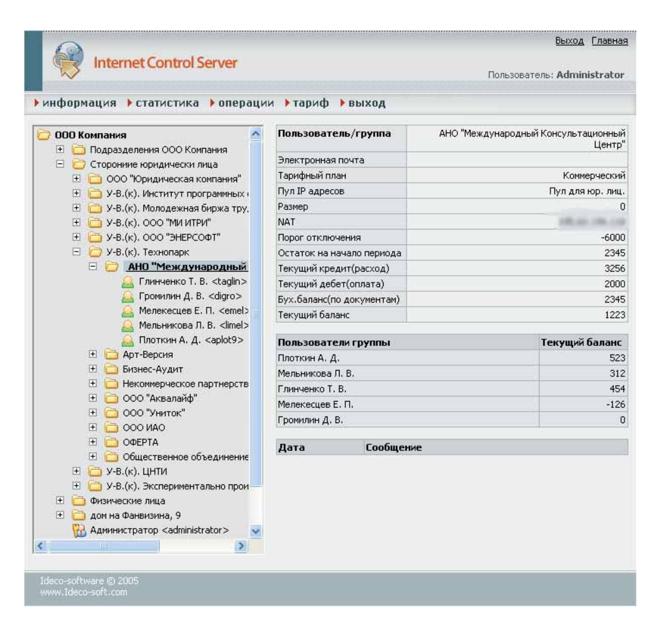
В системе присутствует интеллектуальный модуль Quality of Service (QoS) для автоматического определения приоритета трафика.

Для того, чтобы использовать очереди на входящий трафик, на ядро были наложены дополнительные патчи. По словам разработчиков, из всех протестированных патчей наиболее результативным показал себя IMQ, который создает hook в *iptables*-таблице mangle. Однако, IMQ работает только при скоростях не более 10 Мбит, и, чтобы очереди на базе IMQ работали при больших скоростях, необходимо перевести ядро в режим 1000Гц, а *netsheduler* – в режим отсчета тиков от RTC-таймера вместо jiffies.



Интеллектуальная приоритезация осуществляется маркировкой пакетов в зависимости от скорости соединения и протокола. Когда пакет попадает в очередь HTB устройства IMQ, то на основе маркировки он помещается в свой класс, т.е. своего рода подочередь. Соответственно, пакеты высокоскоростных прожорливых соединений, например, скачивание файлов, не задавят более важные пакеты, например, ssh. Кроме этого, используется очередь sfq, и пакеты разных соединений, находящихся в очереди, выходят из нее по принципу рулетки, что позволяет равномерно распределить канал между пользователями и между приложениями.

В *Ideco ICS* имеются и дополнительные сервисы: почтовый сервер Postfix с антивирусом ClamAV, web-сервер с поддержкой PHP и *MySQL*. Также присутствует FTP-сервер Vsftpd и «служба точного времени» ntpd. Все службы запускаются в своих chroot каталогах.



Кто-то скажет, что все возможности *Ideco ICS* можно реализовать и на обычных дистрибутивах Linux. Это действительно так, и при должной квалификации персонала задача реализуема, но сколько она отнимет сил и времени? В результате будет получено, хотя и оптимизированное под нужды конкретного предприятия, но все же единичное решение – и в другой раз, возможно, придется все начинать с нуля. *Ideco ICS*, напротив, типовое решение, проверенное и работающее во многих организациях.

Какая польза сообществу Open Source от подобных коммерческих проектов? В результате внимательного тестирования Linux и использования его в типовых решениях выявляются уязвимости и ошибки. Исправления для них передаются разработчикам соответствующих открытых проектов. Кроме этого, проводится своего рода «селекционный отбор», и в коммерческие решения попадают только самые проверенные и зрелые разработки. Таким образом, это не просто эксплуатация Linux в коммерческих продуктах, это тесное сотрудничество, позволяющее совершенствовать Open Source и продвигать Linux на корпоративный рынок. **LXF**

Web-сервер Apache для начинающих web-мастеров и администраторов



APACHE – друг «бледнолицых»

ЧАСТЬ 2: Если бы *Apache* умел только отдавать статические html-файлы, разве бы он был столь популярен? К счастью, *Apache* способен на гораздо большее... Сегодня **Сергей Супрунов** научит вас поддерживать динамические web-сайты

Как упоминалось в первой части, *Apache* разработан по модульному принципу. Модули позволяют наращивать функциональность практически до бесконечности. Во многие дистрибутивы включается до 40–50 модулей, покрывающих основные потребности, но их число этим, естественно, не ограничивается. Модули могут быть либо вкомпилированы в основной двоичный файл (**httpd**, **Apache**, **Apache2**, в зависимости от дистрибутива), либо собраны для динамического подключения. Список первых можно получить по команде **httpd -l** (опять-таки с поправкой на имя), вторые могут располагаться либо в базовом каталоге web-сервера в **modules**, либо в **/etc/Apache2/mods-available**, **/usr/lib/Apache2/modules** и т.п. Обычно список имеющихся в дистрибутиве модулей можно узнать из конфигурационного файла *Apache* по строкам LoadModule.

Например, рассмотренная в прошлый раз возможность размещать странички пользователей в их домашних каталогах обеспечивается модулем *mod_userdir*. За автоматическое построение индекса каталога отвечает mod_autoindex (зачастую компилируется вместе с сервером). Заметьте, что если вы хотите динамически подгружать модули, *Apache* должен быть статически собран с модулем *mod_so* – именно он отвечает за подключение дополнительных модулей «на лету».

Ещё один часто используемый и потому обычно собираемый статически модуль – *mod_alias*. Его зона ответственности, как следует из названия – псевдонимы для тех или иных каталогов. Например, вы можете указать:

```
Alias /files /mnt/cdrom
```

Теперь при запросе http://www.yourserver.ru/files/my.mp3 web-сервер будет искать нужный файл в каталоге **/mnt/cdrom**. Не забывайте только, что первым параметром указывается путь от корня сайта, вторым – от корня файловой системы.

О некоторых других модулях коротко сказано во врезке, а подробности ищите в документации (начните поиск со страницы http://httpd.Apache.org/docs/2.2/mod/).

Месяц назад Мы научились использовать *Apache* для поддержки статических web-сайтов.

Будьте динамичнее: основы CGI

До этого мы работали только со статическими страницами. Однако это лишь небольшой процент тех возможностей, которые предоставляют нам web-технологии. Сайт, не позволяющий оставить своё мнение в гостевой книге или хотя бы просто зарегистрироваться, чтобы при следующем посещении получить адресованное лично вам «Здрассте» смотрится в современном мире по меньшей мере скучно.

Одной из первых технологий, которые наделяют сайт способностью к взаимодействию с пользователем и позволяют динамически создавать содержимое в зависимости от тех или иных условий, является Common Gateway Interface (CGI). CGI позволяет серверу не просто «выплеснуть» в сокет содержимое файла, а запустить исполняемую программу и вернуть клиенту её вывод. Реализован этот функционал в модуле *mod_cgi*.

Например, клиент запрашивает ресурс http://www.server.ru/cgi-bin/gbook.cgi. Если в настройках сервера каталог **cgi-bin** определён как предназначенный для запуска CGI-сценариев, то сервер запустит файл **gbook.cgi**, передав через переменные окружения и входной поток информацию из клиентского запроса. Сценарий выдаст в свой стандартный поток **STDOUT** информацию (как правило, html-страницу), которую сервер перешлёт клиенту.

По умолчанию (практически во всех дистрибутивах), *Apache* уже настроен на выполнение CGI-сценариев в некотором особом каталоге. Имя этого каталога вы найдёте в конфигурационном файле в строке директивы **ScriptAlias** (в Ubuntu это **/usr/lib/cgi-bin**). Любой файл из этого каталога, к которому выполняется запрос, *Apache* запускает в специально созданном для этого окружении (сам сценарий должен иметь права на выполнение), а всё, что сценарий выведет операторами print (или аналогичными), будет отдано клиенту.

Для каталога, указанного в **ScriptAlias**, должна быть задана опция **ExecCGI** (директивой **Options +ExecCGI**). При желании обрабатывать CGI-сценарии вне данного каталога, можно сопоставить соответствующий обработчик с тем или иным расширением:

```
AddHandler cgi-script .cgi
```

Есть ещё одна директива, включающая обработку CGI – *SetHandler* cgi-script. Будучи указанной в одной из секций, она заставит *Apache* рассматривать все подпадающие под действие секции файлы в качестве CGI-скриптов. Фактически, упомянутая выше директива ScriptAlias аналогична следующим строкам:

```
Alias /cgi-bin /usr/lib/cgi-bin
<Directory /usr/lib/cgi-bin>
  SetHandler cgi-script
</Directory>
```

По соображениям удобства сопровождения [и безопасности, – прим. ред.], не рекомендуется размещать CGI-сценарии за пределами отдельного каталога, выделенного специально для них.

Для разработки CGI-сценариев вы можете использовать любой удобный вам язык программирования – нужно только, чтобы система умела работать с ним, а он сам – поддерживал минимум необходимых средств (например, чтение данных из входного потока и переменных окружения).

Листинг test.fcgi

```
#!/usr/bin/perl

use FCGI;

$cnt = 0;
while(FCGI::accept >= 0) {
  $cnt++;
  print "Content-Type: text/html\n\n";
  print "Это ${cnt}-й запрос ресурса";
}
```

Ещё быстрее: FastCGI

Технология CGI очень хороша своей универсальностью и гибкостью, но имеет один существенный недостаток, особенно сильно сказывающийся при работе с языками сценариев типа Perl, Python, Ruby – низкое быстродействие. Ведь при каждом запросе будет запускаться интерпретатор соответствующего языка, выполняться компиляция текста сценария в байт-код и только после этого – выполняться сценарий. Затем все эти ресурсы нужно ещё и высвободить. В режиме web-сервера такие накладные расходы могут доходить до 80–90% от общих затрат времени на обслуживание запроса [это становится особенно актуально для инструментариев типа Ruby On Rails, – прим. ред.].

Для решения этой проблемы был разработан интерфейс FastCGI. В отличие от CGI, здесь сценарий постоянно находится в памяти (т.е. запуск интерпретатора и компиляция требуются только при первом запуске; есть возможность настроить предварительную компиляцию при старте сервера), а вместо переменных окружения и потоков ввода-вывода взаимодействие с web-сервером осуществляется через двунаправленное соединение.

Чтобы это работало, сценарий должен быть написан соответствующим образом (в частности, не закрывать соединение после выполнения, а возвращаться к ожиданию следующего запроса).

Для работы с FastCGI нужно, во-первых, установить модуль *mod_fastcgi*. Если ваш менеджер пакетов ничего о нем не знает, это можно сделать и вручную, скачав архив с сайта www.fastcgi.com. Следуйте инструкциям в **README** и **INSTALL** (для *Apache* 2.0 – **INSTALL.AP2**). Для успешной сборки вам понадобятся исходные коды *Apache*. Если у вас *Apache 2.2*, то придётся наложить патч, например, отсюда: http://www.fastcgi.com/archives/fastcgi-developers/2005-December/004060.html.

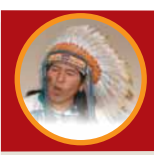
Во-вторых, в конфигурации *Apache* нужно указать такие директивы:

```
LoadModule fastcgi_module modules/mod_fastcgi.so
<IfModule mod_fastcgi.c>
   AddHandler fastcgi-script .fcgi
   SocketPath /var/lib/apache2/fcgid/sock
</IfModule>
```

Не забудьте исправить пути на свои. В данном примере с файлами, имеющими расширение **fcgi**, будет сопоставлен обработчик *fastcgi-script*. Теперь можно создать сценарий, подобный **test.fcgi**, приведенному во врезке слева.

Неоднократно запуская этот сценарий, вы убедитесь, что он постоянно присутствует в памяти – переменная **$cnt** будет наращиваться с каждым новым запросом. Обратите внимание на обязательность включения модуля FCGI.

Нужно заметить, что поскольку FastCGI-сценарий не выгружается из памяти, к его качеству предъявляются гораздо большие требования. Например, утечка памяти из-за небрежно написанного кода, совершенно не критичная для обычного CGI-скрипта, может привести к захвату всех доступных ресурсов и фактическому блокированию дальнейшей работы. Также следует избегать глобальных переменных, так что будьте внимательны. »



Коротко о модулях

mod_asis – позволяет отправлять клиенту файлы в «сыром» виде, без добавления каких-либо заголовков (полезен, если вы хотите сами сформировать нестандартные заголовки).

mod_dav – поддержка протокола WebDAV, позволяющего работать с файлами по протоколу HTTP.

mod_imap – поддерживает работу «серверных» карт изображения (Image Map), когда щелчок пользователя по той или иной области рисунка обрабатывается сервером по-своему.

mod_rewrite – предоставляет широчайшие возможности по перезаписи URL-адресов; например, вы можете перенаправлять запросы на другие ресурсы, используя регулярные выражения для определения правил подстановки.

mod_speling – автоматически исправляет ошибки в имени ресурса, подбирая наиболее подходящие варианты (например, если файла **inex.html** нет, пользователю будет возвращён **index.html**).

mod_ssl – обеспечивает работу по защищённому протоколу SSL.

mod_suexec – позволяет выполнять CGI-сценарии с правами другого пользователя, а не того, от имени которого работает *Apache*.

mod_vhost_alias – позволяет задавать «шаблоны» виртуальных хостов, что полезно при настройке большого числа виртуальных сайтов со схожими параметрами.

Server Side Includes



Если возможности CGI для решения ваших задач избыточны, можно использовать механизм «серверных включений» – SSI (Server Side Includes). Заключается он в том, что в html-код включаются специальные ssi-директивы, а *Apache*, прежде чем выдать страницу клиенту, заменяет эти директивы результатом их выполнения. Для использования SSI нужно разрешить этот механизм с помощью Options +Includes (IncludesNoExec для большего уровня безопасности, без права запуска внешних программ) и установить выходной фильтр для расширений, которые должны обрабатываться, с помощью AddOutputFilter Includes .shtml (можно использовать SetOutputFilter Includes для «привязки» фильтра ко всем файлам текущей секции). В *Apache 1.3* используется директива AddHandler server-parsed .shtml.

Помимо «официальной» реализации – *mod_fastcgi* – можно найти и другие. Например, в репозиториях Ubuntu доступен *mod_fcgid*, совместимый с *mod_fastcgi*.

Модули специального назначения

Второй путь преодолеть проблемы производительности CGI – специализированные модули, взаимодействующие с *Apache* API и отвечающие за выполнение сценариев на том или ином языке. К таковым относятся *mod_perl*, *mod_python*, *mod_php* и т.д. Здесь уже сам модуль выполняет интерпретацию сценария (т.е. отдельно запускать Perl или Python не требуется), а за счёт кэширования единожды откомпилированный сценарий может быть сохранён в памяти и будет находиться в полной боевой готовности. При этом со стороны сценария не требуется какой-то специальный код, и во многих случаях в среде *mod_** без проблем будут исполняться и обычные CGI-скрипты (хотя определённые ограничения есть и здесь – например, в случае mod_perl нельзя использовать директивы **require**, **_DATA_** и **_END_**, функцию **exit()**; требуется также осторожность при работе с глобальными переменными, поскольку они сохраняют свои значения между запусками).

Но платить за это приходится потерей универсальности – очевидно, что *mod_python* не сможет обработать ваш Perl-скрипт, а для менее распространённых языков может вообще не оказаться модуля.

Эти модули настраиваются аналогично рассмотренным выше – либо с помощью **AddHandler** по расширению, либо – **SetHandler** для всех файлов в секции. Естественно, предварительно модуль должен быть подключен с помощью **LoadModule**. Каждый модуль предоставляет и дополнительные параметры, позволяющие более гибко использовать его возможности. Например, mod_perl позволяет задать тип обработчика. Строка **PerlHandler Apache::Registry** обеспечивает однократную компиляцию сценариев, которые в дальнейшем постоянно присутствуют в памяти. **PerlHandler Apache::PerlRun** включит другой обработчик, который позволяет исполнять даже CGI-сценарии, не работающие в окружении **Apache::Registry**, например, активно использующие глобальные переменные, поскольку в этом случае сценарий каждый раз выгружается из памяти после исполнения (единственное преимущество по сравнению с CGI здесь в том, что не требуется обращаться к внешнему интерпретатору).

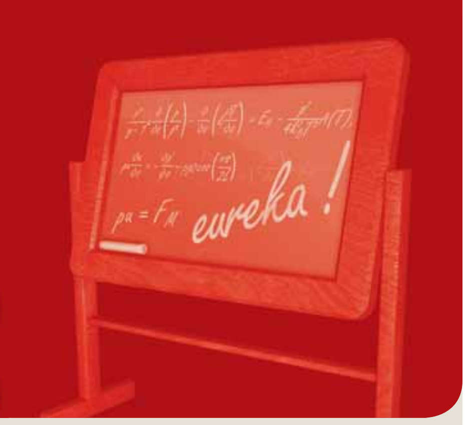
Иногда обработчик привязывается к расширению через MIME-тип, что довольно часто практикуют для подключения *mod_php*:

```
LoadModule php5_module /usr/lib/apache2/modules/
libphp5.so
<IfModule mod_php5.c>
  AddType application/x-httpd-php .php .phtml .php3
  AddType application/x-httpd-php-source .phps
</IfModule>
```

Подробности ищите в документации по конкретному модулю.

На этом мы завершим наш краткий экскурс в *Apache*. Возможностей у этого сервера, как видите, множество. Если вы желаете поглубже разобраться в них, то более подробное изучение можно начать с сайта http://httpd.apache.org. Ну и Google тоже не зря столь активно использует открытые проекты... LXF

Учебники



Наши эксперты помогут вам с любым приложением Linux



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Linux в каждый дом и в каждый карман

Прямо чудо какое-то. Почти одновременно две никак не связанные друг с другом компании решили напомнить о себе, создав информационный повод. Фирма Loveland (CO) — Terra Soft Solutions сообщила, что разрабатываемый ею дистрибутив Yellow Dog Linux 5.0 будет поддерживать Sony Playstation 3 прямо из коробки. Компания ACCESS Co. Ltd. и ее подразделение PalmSource представили ACCESS Linux Platform для наладонных компьютеров.

«Пока под Linux нет игр — ему ничего не светит...»
«Пока нет Linux-телефонов — ему ничего не светит...»
anonymous, разновидность коллективного разума на linux.org.ru

Примечательно, что обе упомянутые компании переживают не самые лучшие времена. Yellow Dog Linux специализировалась на PowerPC/Macintosh и переход Apple на x86 серьезно спутал ей карты. Хозяева PalmSource оказались перед фактом обвала спроса на Palm OS — операционной системы, которая всего несколько лет назад являлось синонимом «системы для КПК».

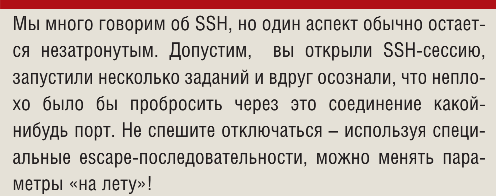
И для тех, и для других выиграть – это шанс не просто улучшить свои дела, а выдвинуться на первые роли в кошмарно изменчивом рынке высоких технологий.

А что от этого «перепадет» сообществу? Очевидно, что больше никто не будет говорить, что под Linux нет игр. Если ACCESS Linux будет иметь пользовательскую функциональность, сравнимую со старым добрым Palm OS, то я знаю, что будет стоять на моем следующем наладоннике.

Будущее неизбежно внесет свои коррективы, но помечтать никто не запрещает.

E.M.Baldin@inp.nsk.su

В этом выпуске...



СОВЕТ МЕСЯЦА: Побег из SSH

Мы много говорим об SSH, но один аспект обычно остается незатронутым. Допустим, вы открыли SSH-сессию, запустили несколько заданий и вдруг осознали, что неплохо было бы пробросить через это соединение какой-нибудь порт. Не спешите отключаться – используя специальные escape-последовательности, можно менять параметры «на лету»!

Escape-последовательность – это не более чем набор символов, приказывающий утилите, с которой вы работаете (в данном случае, SSH) «отвлечься» и выполнить какую-то специальную задачу. Вы наверняка встречались с escape-последовательностями при работе в оболочке.

Самая полезная escape-последовательность SSH – это тильда (~), за которой следует заглавная буква C (Shift+c). Пока вы вводите эту последовательность, никаких изменений не будет, зато потом приглашение изменится на ssh>. Это значит, что вы перешли в командную строку SSH, где можно соединить порты удаленной и локальной машин и перебросить данные через защищенное соединение.

Мы использовали этот прием в LXF71, чтобы организовать туннель между локальным компьютером и сервером Squid – для этого использовался аргумент -L. Он же поможет нам и сейчас. Набрав -L8090:localhost:3128, вы создадите туннель, не перезапуская SSH-сессию. Для проброса портов можно также использовать escape-последовательность ~#, а отменить все пробросы можно с помощью -Krhostport. Например, чтобы закрыть созданный ранее туннель, наберите -KR3128.



Первые шаги Советы для тех, кто только начинает знакомиться с Linux

OOo Base: Лоск

Данные бесполезны, если невозможно найти необходимую информацию.





Наш эксперт

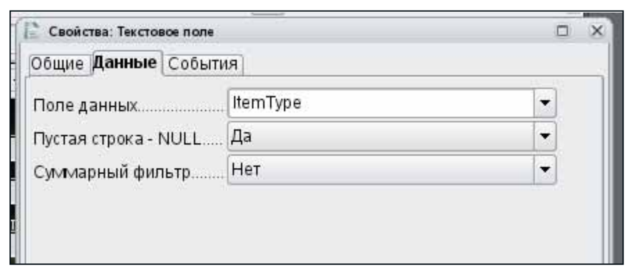
Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется со времен Dragon 32.

В прошлом номере я явил миру свои привычки по закупке товаров, с целью показать вам, как строится простая база данных домашнего имущества. Напомню, что мы использовали мастера *OpenOffice.org Base*, чтобы определить, с какими данными работать (мастер таблиц), и создать простую форму для ввода данных (мастер форм); затем использовали табличный интерфейс для ввода, сортировки и фильтрации данных. В итоге получилась одна таблица и одна форма. Наша таблица имела несколько «полей», то есть заголовков столбцов: например, Item (Вещь) и Serial Number (Серийный номер), и мы вручную ввели некоторые данные в ячейки. Теперь добавим в таблицу место для заметок и категорий, поиграем с интерактивными виджетами форм и изучим запросы и отчеты как средство реорганизации базы данных.

Наша новая работа – редактирование таблиц базы данных с целью добавить место для заметок и опции ItemType (тип предмета, принадлежность): последняя поможет пользователю находить все предметы, относящиеся, скажем, к «компьютеру» или «кухне». Итак, запустите *OOo Base* и откройте созданную вами базу. Перейдите к таблицам, щелкнув на соответствующем пункте в левой панели главного окна *Base*, а затем правой кнопкой мыши на таблице, которую мы создали.

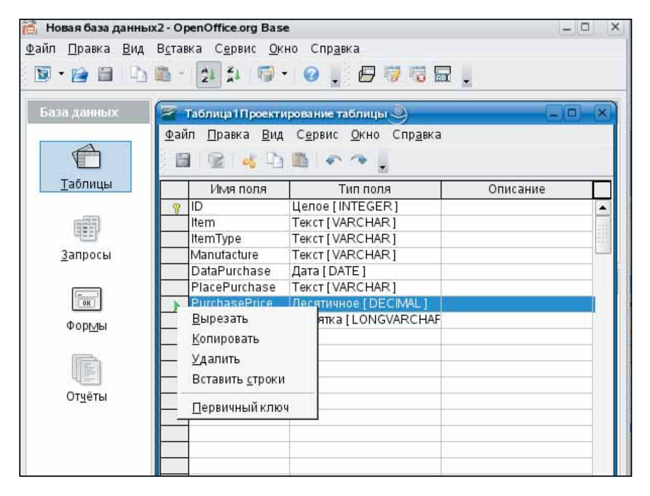
Теперь выберите пункт контекстного меню Edit (Правка), для перехода в экран редактирования таблицы. Этот экран, как и сама таблица, имеет типичный вид электронной таблицы, но сейчас в нем только три столбца с именами Field Name (Имя Поля), Field Type (Тип Поля) и Description (Описание) (см. рисунок внизу). Столбцы Name и Type должны быть заполнены полями, которые вы определили при помощи мастера таблиц в прошлый раз. Для добавления нового поля данных необходимо просто щелкнуть на вакантной ячейке столбца Имя поля, задать имя и определить корректный тип данных. Новому полю ItemType можно придать стандартный тип Текст [VARCHAR], но новое поле Notes (Описание) в принципе может содержать много информации, поэтому лучше использовать для него тип данных Memo (Памятка) [LONGVARCHAR]. Не пугайтесь: элемент VARCHAR означает просто «изменяемый символ», а LONGVARCHAR – то же самое, только подразумевает более длинные текстовые строки.

> «Благодаря формам, отображение данных удобно для восприятия.»



› **Щелчок по любому элементу в форме вашей базы данных откроет диалог настройки свойств.**

Снова посмотрите на нижний рисунок. Если вы хотите изменить порядок полей в таблице, просто переместить поля вверх или вниз в списке нельзя – зато можно вырезать и вставлять строки, щелкая в крайней левой ячейке, где появляется маленький зеленый треугольник [столбец без надписей, с серым фоном – прим. пер.], а затем щелкнуть правой кнопкой для вызова контекстного меню. Когда вы сделаете это, все строки ниже вырезанных сдвинутся вверх к первой свободной строке, и любая операция вставки из буфера разместит вырезанную строку(-ки) внизу таблицы. На вид сложно, но вы быстро привыкне-



› **Правый щелчок на крайней левой колонке выведет контекстное меню со стандартными опциями.**

В прошлом месяце Мы построили базу данных при помощи *OpenOffice.org Base*.

вашей базы данных

Во второй части статьи о базах данных, **Энди Ченнел** облагораживает формы и отчеты.

те к этому! Добавив новые поля и разместив их в нужном месте, вы можете – при желании – ввести описание, затем щелкнуть в верхней части окна на иконке панели инструментов с изображением дискеты для сохранения таблицы.

Закрыв окно правки, дважды щелкните на таблице в главном окне *OOo Base*, чтобы открыть ее в режиме добавления данных. Заметьте, что если у вас в таблице уже были данные, то они не уничтожаются при редактировании структуры таблицы; но вам потребуется пройтись по старым записям и заполнить прибавившиеся поля – всегда лучше делать все правильно с самого начала.

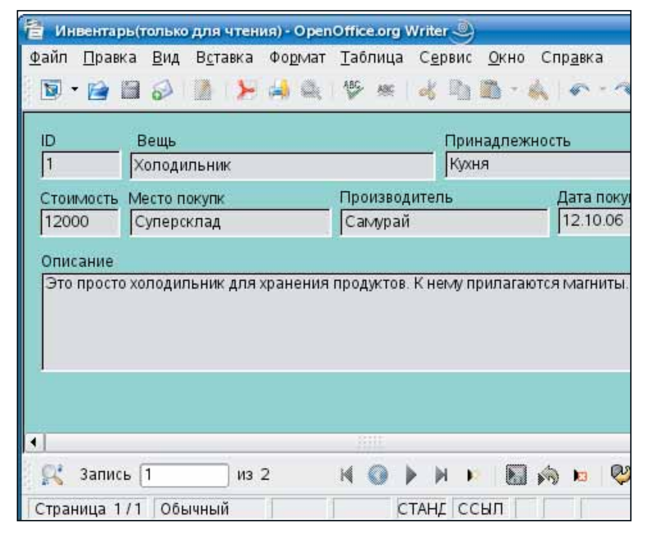
Изменение формы

Хотя для ввода данных можно обойтись и таблицей, намного приятнее делать это в созданной нами форме (см. рисунок справа). Формы также облегчают восприятие отдельных элементов, потому что каждая запись отображается в своем собственном окне. В прошлый раз мы строили форму при помощи Мастера форм, а теперь добавим Notes (Описание) и ItemType (Принадлежность), которые мы уже добавили в таблицу; также улучшим дизайн, добавив виджет даты.

Для начала, щелкните на иконке Формы слева и затем выполните правый щелчок на форме Инвентарь (Household Inventory) (или как вы ее там назвали) и выберите Edit (Правка) для вызова главного окна редактирования. Если панелей инструментов Form Design (Разработка формы) и Form Controls (Элементы Управления) на экране нет, активируйте их из меню View > Toolbars (Вид > Панели инструментов).

Первое, что следует сделать – расчистить пространство для нового поля ItemType. К счастью, мастер форм в OOo автоматически группирует поля с их метками, то есть нужно просто щелкнуть на поле и перенести его в новое место. Если вы хотите передвигать метки и поля отдельно, разгруппируйте их, выбрав желаемое поле и выполнив Format > Group > Ungroup (Формат > Группировать > Разгруппировать) – используйте метод «щелчок, Shift+щелчок» для выбора нескольких элементов на странице, а затем путем Format > Group > Group (Формат > Группировать > Группировать) вновь объедините их. Мы щелкали на элементах и переносили их так, чтобы расчистить место в правом верхнем углу формы для размещения поля ItemType.

Теперь нам необходимо вставить в форму новую строку ввода для ItemType. На панели инструментов Forms Controls (Элементы управления) выберите иконку Text Box (Текстовое поле) и нарисуйте новое »



› **Ввод данных через форму удобнее использования таблицы и облегчает просмотр деталей записи.**

Скорая помощь

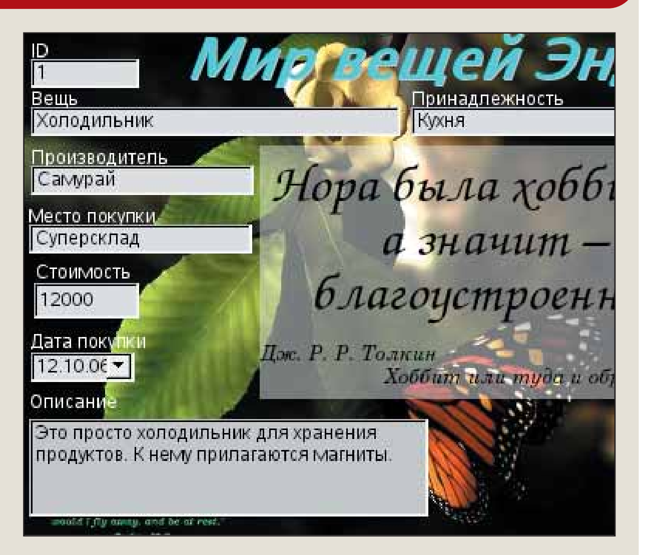
Вы всегда можете найти последнюю версию *OOo* на www.openoffice.org. Если вы работаете с версией, поставляемой с вашим дистрибутивом, то, вероятно, пора выполнить обновление...

Красивым быть не запретишь!

Если вы присмотритесь к дизайнеру форм *OOo Base*, то заметите, что на самом деле это просто *OOo Writer*. И это здорово: значит, все инструменты текстового процессора доступны для разработки прекрасных форм. В принципе, формы должны быть серыми, нейтральными и практичными – таков уж британский подход; но любой последователь Джона Китса понимает, что мы ищем правду в красоте, и в красоте – правда. Давайте сделаем красиво.

Для начала, откройте ранее созданную форму, щелкнув на её имени правой кнопкой и выбрав Edit (Правка). Мы имеем тюремного вида фон под названием Teal (зеленовато-голубой), и от него я собираюсь избавиться в первую очередь. Выберите Формат > Страница, затем перейдите на вкладку Фон. Здесь можно изменить цвет страницы или, еще лучше, поместить в качестве фона приятную картинку: в выпадающем списке Тип выберите Графический объект, нажмите кнопку Обзор и подберите подходящее изображение. В нижней части окна располагаются управляющие опции, позволяющие растянуть картинку на всю страницу, расположить в виде мозаики или поместить в конкретной позиции.

Некоторые изображения не подходят для использования в качестве фона, но вы можете осветлить их в *Gimp*, чтобы текст в элементах формы оставался видимым. Вы можете также слегка передвинуть элементы, чтобы расположить их более эстетично, разместить прозрачный текст с подходящей философской сентенцией или изменить заголовок формы на менее официальный.



› **Не все формы такие угрюмые, как Ф286. Нашей базе данных явно к лицу бабочки, смешные шрифты и глубокомысленные цитаты. Или нет?**

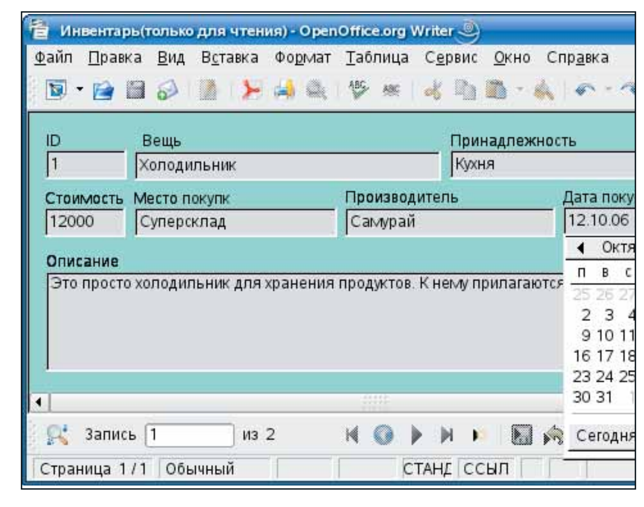
» текстовое поле справа – точно так же, как рисуется прямоугольник в графическом пакете, то есть можно изменять размер, щелкнув на элементе и перетаскивая любую управляющую точку по углам и краям элемента. Если вам необходима большая точность, чем позволяют ваши глаза, щелкните правой кнопкой на элементе и выберите в контекстном меню пункт Position and Size (Положение и размер). В появившемся окне можно менять – угадайте, что? – положение и размер выбранного элемента.

После размещения и установки размера элемента можно определить, как база данных будет его «видеть». Дважды щелкните на элементе и в появившемся диалоговом окне Properties (Свойства) выберите вкладку Data (Данные). Главная опция здесь Datafield (Поле данных), и его действительно необходимо связать с заголовком таблицы базы данных, в данном случае это ItemType (Принадлежность). Раз уж мы сюда попали, осмотримся: здесь можно много чего понаделать. Например, на вкладке General (Общие) располагаются опции для добавления 3D-вида формы (как на рисунках этого урока); можно изменить порядок обхода элементов (то есть порядок, в котором пользователь перемещается между элементами по клавише Tab); и установить текст по умолчанию для каждого элемента или перевести элемент в режим только для чтения. На вкладке Events (события) можно связать несколько макросов с различными действиями мыши и клавиатуры – но это уже вне рамок данного урока.

Настроив элемент по своему вкусу, выберите Label Field (Метка), нарисуйте площадку выше (или сбоку) предварительно сделанного текстового поля, затем введите желаемый текст метки – я ввел текст ItemType для простоты. Сделав это, щелкните на элементе, затем при нажатой клавише Shift на обоих элементах, и сгруппируйте их. Так же

Скорая помощь

OOo Base предоставляет многострочные компоненты TextBox (Текстовое поле) со всеми возможностями форматирования, это удобно для больших заметок. Однако это немного «сырая» опция, советую быть с ней поосторожнее.



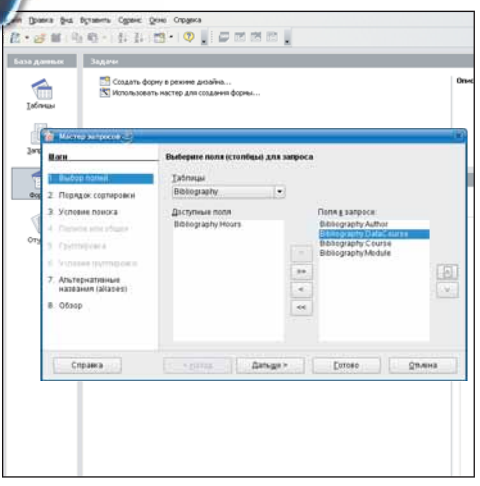
› **Виджет-календарь намного упрощает ввод дат.**

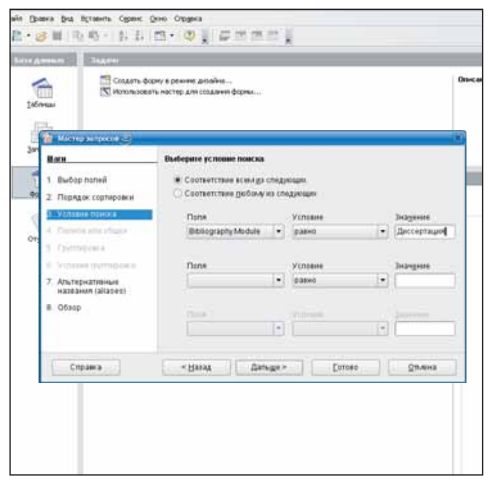
мы поступим с элементом Notes (Описание), но в этот раз сделаем текстовое поле побольше (здесь, вероятно, будет больше информации) и в диалоговом окне Properties (Свойства), на вкладке General, выберите для опции Text type (Тип текста) значение Multi-Line (Многострочный).

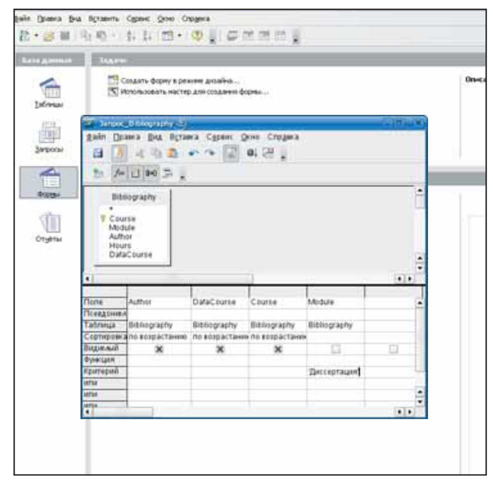
Пока мы в Редакторе форм, можно также добавить виджет-календарь для поля Date Purchased (Дата покупки), это намного упростит заполнение формы. Дважды щелкните на поле и на вкладке General найдите строку с названием Dropdown (Выпадающий); выберите Yes (Да). Вот и все. Нажмите кнопку Save (Сохранить), и все готово для



Шаг за шагом: Запуск запроса в OOo Base







1 Определите условия поиска

Первый шаг при выполнении запроса к базе данных – выбор опции Query (Запрос) в левой части главного окна, а затем выбор опции Use Wizard To Create Query... (Использовать мастер для создания запроса...) для запуска знакомого мастера. Главное – определить, какой конкретный элемент базы данных вы хотите отобразить. Щелкните на кнопке Next (Далее), затем выберите способ упорядочения результатов – в моем примере с библиографией я выбрал сортировку по автору, затем по году и затем по названию. Нажмите кнопку Next (Далее) вновь для перехода в раздел Search Conditions (Условия Поиска). Вот это очень важно, поскольку позволяет включить или исключить конкретные записи, и, к счастью, в условиях поиска не обязательно перечислять все поля, выбранные для отображения.

2 Наложение условий на поля

Моя база данных-библиография содержит ссылки на все курсы, который я посещал за последние два года, но меня сейчас интересуют те, что использовались для моей диссертации. В этом случае я могу заполнить опции Field (Поля), Condition (Условие) и Value (Значение) значениями *Books.Module*, *Is Equal To* (*равно*) и *Dissertation* (*Диссертация*) соответственно, что выведет только результаты, которые связаны с Диссертацией в поле Модуль. В последнем диалоговом окне выбираем опцию Modify Query (Изменить запрос), чтобы открыть запрос в Дизайнере до его запуска. Дизайнер имеет две основные панели для показа каждой активной таблицы и критериев запроса, а также обычный набор панелей инструментов и меню. Я просто хочу снять отметку со столбца Module в нижней панели, чтобы имя модуля не отображалось в конечном результате.

3 Выполнение запроса

Вы можете протестировать запрос, нажав иконку Run Query (Выполнить запрос) на панели инструментов, и результирующая таблица отобразится в третьей панели окна. Запросы могут быть запущены в любое время из окна Queries (Запросы); их легко отредактировать, щелкнув правой кнопкой мыши на запросе и выбрав пункт Edit (Правка). Вы, конечно, можете создать сколь угодно большое количество запросов, откуда следует, что на одну и ту же базу данных можно смотреть под разными углами для различных целей.

ввода данных. Можете закрыть окно Дизайнера форм, используя меню File > Quit (Файл > Выход). Вернувшись в главное окно, щелкните дважды на форме Инвентарь, чтобы открыть ее для ввода данных. Немного озадачивает то, что при этом форма базы данных может иметь в строке-заголовке суффикс только для чтения, но не беспокойтесь – записи, вводимые данным способом, сохраняются как обычно. В зависимости от настройки *OOo* на вашей машине, у вас уже может быть доступна правая панель. Если навигационная панель не размещается внизу окна (она содержит кнопки Back (Назад), Forward (Вперед), Save (Сохранить), Undo (Отменить) и т. д.) то для ее вывода выполните View > Toolbars > Form Navigation (Вид > Панели инструментов > Навигация формы).

Начинаем ввод

Мы вводили данные в базу в прошлый раз, через табличный вид, так что несколько записей уже заполнено. Вы можете перемещаться по ним, используя кнопки Back (Назад) и Forward (Вперед) на панели навигации; или сразу начать ввод, нажмите кнопку New Record (Новая запись), расположенную по правому краю первого раздела кнопок (на ней изображена стрелка со звездочкой внутри). Перед вами появится абсолютно пустая запись с мигающим курсором в первом поле, обычно это поле ID. Вы можете начать ввод данных в различные поля, перемещаясь между ними при помощи клавиши Tab в установленном порядке обхода, как указывалось ранее, или выбирая нужное поле мышью. Заполнив запись, опять нажмите кнопку Новая запись. Это сохранит текущую запись и предоставит пустую для ввода данных.

Панель навигации вблизи нижнего края содержит такие инструменты, как фильтрация и сортировка, кнопка отмены, опция Delete Record (Удалить запись), а у правого края панели – иконка для отображения таблицы базы данных и формы, очень полезного для просмотра деталей записей и их связей с другими данными в базе данных.

Превращаем данные в информацию

До сих пор мы работали над структурой базы данных и вводом данных в таблицы. Без этого не обойдешься; но если база данных для чего и нужна, так это для преобразования данных в наглядную информацию; возникает потребность в запросах. Не паникуйте: сами того не подозревая, концепцию запросов вы скорее всего уже освоили, поскольку каждый раз, посещая Google и вводя строку поиска, вы создаете запрос. Интернет можно рассматривать как огромную базу данных, и надежность результатов поиска Google основана на принципе разбиения этой базы по категориям. К примеру, рассмотрим стандартный поиск словосочетания **Linux Format**. На первой странице мы вводим строку поиска **Linux Format** и нажимаем кнопку Поиск. Google на самом деле ищет два слова: **Linux** и **Format**, поэтому страницы, содержащие оба этих слова, будут ближе к началу результирующего списка, чем страницы, содержащие только одно из них. Теперь представьте себе таблицу, используемую для этого поиска. Один столбец может быть озаглавлен Linux, а второй Format, и база данных отдаст приоритет страницам, содержащимся в обоих столбцах [отметим, что это – всего лишь наглядный пример, а не фактическая структура базы данных Google – прим.ред.].

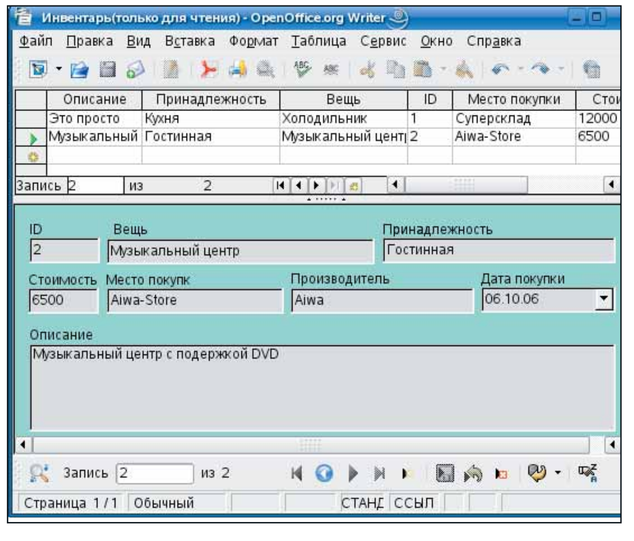
> «Каждый раз, посещая Google и используя поиск, вы создаете запрос.»

В терминах баз данных, запрос – просто способ фильтрации ваших данных для просмотра подмножества базы данных. В прошлый раз мы рассматривали фильтры, и идея запросов в принципе та же – только мы можем сохранить эти критерии и применить их в любое время, или очистить их при необходимости. Рассмотрим пример.

Пример, вместо базы данных Домашний инвентарь (которая маловата для значительных запросов – я обычно закупаю больше!), основан на другом реальном проекте: библиографии книг и журналов, использованных за год обучения в университете. Это применимо к любой базе данных – ваши поля просто будут иметь другие имена. Как это сделать? Прочтите **Шаг за шагом** на предыдущей странице.

Другой стороной поиска являются отчеты. Они выполняют ту же работу, что и запросы, с тем преимуществом, что результаты выводятся в виде таблицы в *OOo Writer*, и их можно распечатать и распространять обычным способом. Процесс создания отчета очень похож на создание запроса, но с дополнительным разделом, где можно выбрать стиль представления данных. Снова, *OOo Base* сохраняет критерии отчета на вкладке Reports (Отчеты), то есть отчет может быть многократно запущен в одной и той же базе данных, если данные изменились.

OOo Base поддерживает большие проекты вроде списка членов и склада инструментов точно так же, как и малые наборы данных. В последних двух выпусках я описал только основы, но эти элементы применяются в большом классе приложений. Всегда помните, что полезность базы данных зависит от грамотности её структуры, так что с самого начала не пожалейте времени на отделку, и поймете, что работать с данными будет куда легче. LXF



› **Незачем скакать от окна к окну – на одном экране видны и форма, и таблица.**

Советы по выделению объектов

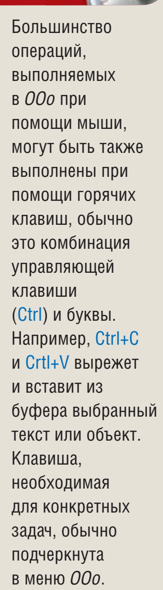
Для рабочего стола Linux и приложений стандартным способом выбора иконок или файлов при помощи мыши является левая кнопка. Как только на элементе произведен щелчок, происходит некое визуальное изменение, сообщающее, что иконка, опция, файл или что там еще выделено: обычно вокруг иконки появляется рамка или меняется цвет фона текстовой метки.

Иногда, однако, необходимо выделить более одного элемента в папке. Чтобы сделать это, можно использовать либо Shift+щелчок, либо Ctrl+щелчок. Первый – для выбора области последовательных файлов, расположенных в файловом менеджере или окне браузера, а второй – для выбора произвольных элементов в списке.

Shift+щелчок подразумевает щелчок кнопки мыши на первом объекте, удерживание клавиши Shift и щелчок на втором объекте. Все, что находится между первым и вторым отмеченными объектами будет выделено и может быть отредактировано, вырезано, копировано или перенесено в другое место. Ctrl+щелчок подразумевает щелчок на первом объекте, удерживание клавиши Ctrl и снова щелчок – таким образом будут выделены только те файлы, по которым вы щелкнули. Можете выбрать сколько угодно файлов.

Скорая помощь





Большинство операций, выполняемых в *OOo* при помощи мыши, могут быть также выполнены при помощи горячих клавиш, обычно это комбинация управляющей клавиши (Ctrl) и буквы. Например, Ctrl+C и Crtl+V вырежет и вставит из буфера выбранный текст или объект. Клавиша, необходимая для конкретных задач, обычно подчеркнута в меню *OOo*.

» **Через месяц** Подобно Google, запустим кое-что из Windows при помощи Wine.

Inkscape: Новое



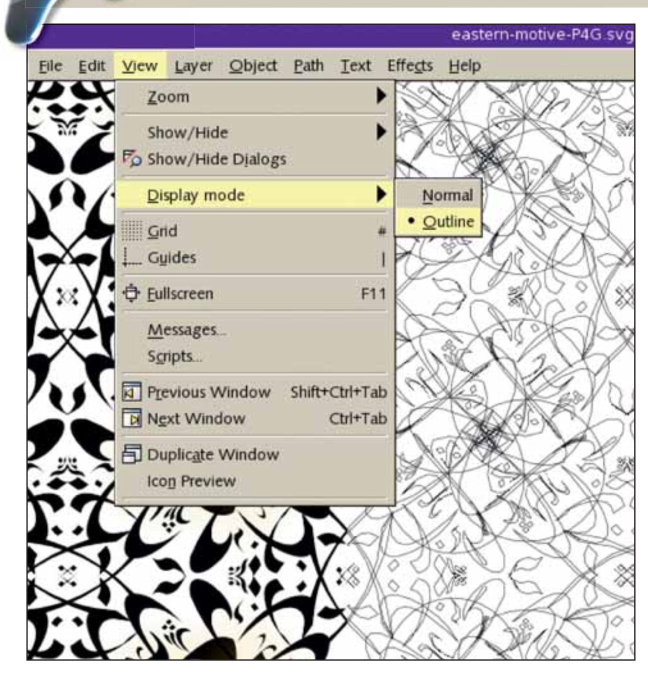
ЧАСТЬ 5: *Inkscape* постоянно совершенствуется – не отставайте! **Дмитрий Кирсанов** продемонстрирует вам новейшие функции программы.

Даже в мире открытого ПО, где перемены являются нормой, стремительные темпы развития *Inkscape* удивляют. Новая версия выходит каждые несколько месяцев, и даже если номер отличается всего на 0.01, происходят заметные перемены по всем направлениям. Такой продуктивности есть несколько причин. Во-первых, *Inkscape* – бесспорный лидер в области векторной графики мира Linux. Естественно, это привлекает многих разработчиков, интересующихся данной темой. Именно разработчики – движущая сила любого проекта, и понятно, что они, а с ними и тестеры, вкладывают свои усилия в известный продукт. (Лишь недавно у *Inkscape* появился достойный соперник в лице *Xara Xtreme LX*, исходный код которой

Наш эксперт

Дмитрий Кирсанов
Графический дизайнер, специализирующийся на создании логотипов и web-сайтов. Консультант, пишет книги и статьи о дизайне и XML-технологиях.

Шаг за шагом: Новые инструменты Inkscape



1 Производительность и режим «Силуэт»

В своих ранних версиях *Inkscape* считалась довольно быстрой программой, но с течением времени пользователи все чаще просили увеличить скорость работы. Неудивительно, что оптимизация и стала главной заботой разработчиков версии 0.44. Скорость обновления экрана и особенно редактирования узлов заметно повысилась. Если для вашего документа и этого мало, попробуйте режим Силуэт (Outline): все объекты отобразятся в виде контуров толщиной в один пиксель, что по определению повышает скорость.

2 Индикатор текущего стиля

Слева от строки состояния появился новый элемент управления, отображающий текущий стиль (цвета заливки и штрихов, ширину штрихов и прозрачность) выделенного объекта (A). Это весьма гибкий инструмент управления, намного ускоряющий работу в *Inkscape*. Его горячие клавиши и контекстное меню упрощают доступ к основным операциям со стилями, без обращения к диалогу Заливка и Штрих (Fill and Stroke) (B).

Месяц назад С помощью Inkscape мы создали элегантную web-страницу, готовую для перевода в HTML.

в версии 0.44



был опубликован в марте этого года. Мы сравним эти программы на следующем занятии.)

Во-вторых, даже на фоне открытых проектов *Inkscape* отличается необычайной открытостью и дружелюбием. Получить доступ для записи в репозиторий исходного кода совсем не трудно, а «старожилы» списка рассылки привечают и ободряют новичков. Важно и то, что все новейшие снимки исходного кода *Inkscape*, как правило, работают без сбоев, поэтому существует довольно крупное сообщество пользователей, отслеживающих *Inkscape* через SVN. Они пользуются программой ежедневно, мгновенно сообщают о сбоях и недостатках и вносят заметный вклад в создание новых функций.

На данном этапе большинство новых функций *Inkscape* только помогают идти вровень с именитыми коммерческими соперниками. Но команда *Inkscape* полна решимости сделать лучший векторный редактор для всех платформ, и каждая новая версия привносит уникальные функции, которые выделяют программу среди конкурентов.

Последняя версия, 0.44 [в настоящее время для скачивания доступна версия 0.44.1, собранная в сентябре 2006 г. – прим.ред.], не стала исключением. Фактически, это наиболее крупная версия (одни только комментарии составляют около 10 000 слов!). Тем, кто пользуется *Inkscape* для создания произведений искусства, web-дизайна или же просто экспериментирует, данный урок продемонстрирует наиболее значительные новшества 0.44 (в контексте состояния искусства векторной графики), чтобы облегчить и усовершенствовать вашу работу.

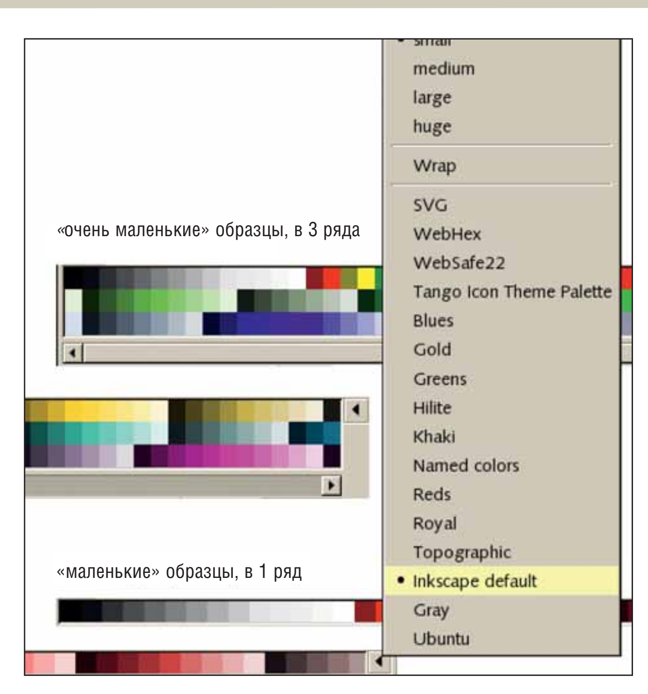
На переднем крае

В векторной графике новые идеи, инструменты и концепции появляются непрерывно. Для тех, кто любит рискнуть, в новейших сборках после 0.44 есть кое-что интересное:

» **Новое диалоговое окно откатов** представляет полный список операций, произведенных вами в течение текущей сессии, позволяющий пересмотреть и отменить любое количество действий разом.

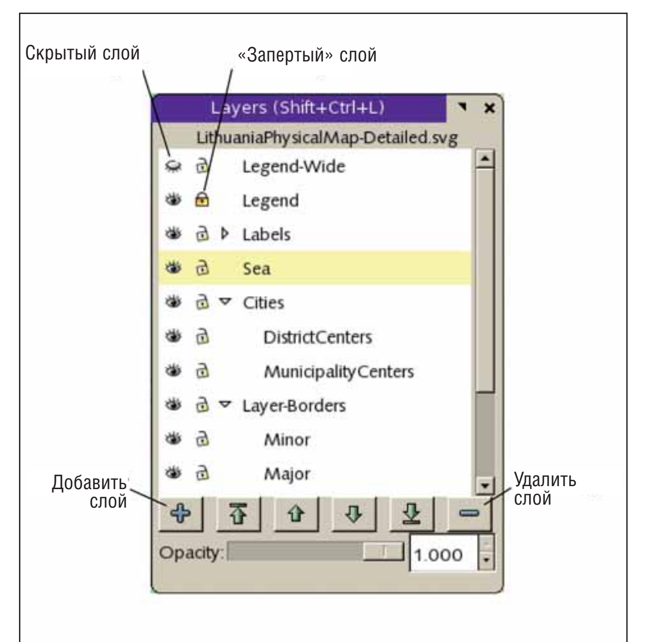
» **Дальнейшее повышение скорости и качества**, чувствительности интерфейса, что делает работу в Inkscape значительно приятнее.

» **Экспорт в PDF** с помощью командной строки.



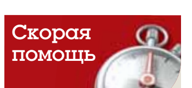
3 Передвижная палитра

В версии 0.43 появилась плавающая палитра, в 0.44 она по умолчанию прикреплена к нижней части окна. Примерно так же палитра выглядит в *Xara Xtreme LX* и *Corel Draw*. Установлена новая палитра по умолчанию, и есть несколько дополнительных (в основном заимствованных у *Gimp*). Палитра может быть показана в виде одного или нескольких рядов цветовых образцов размером от «мелкого» до «крупного».



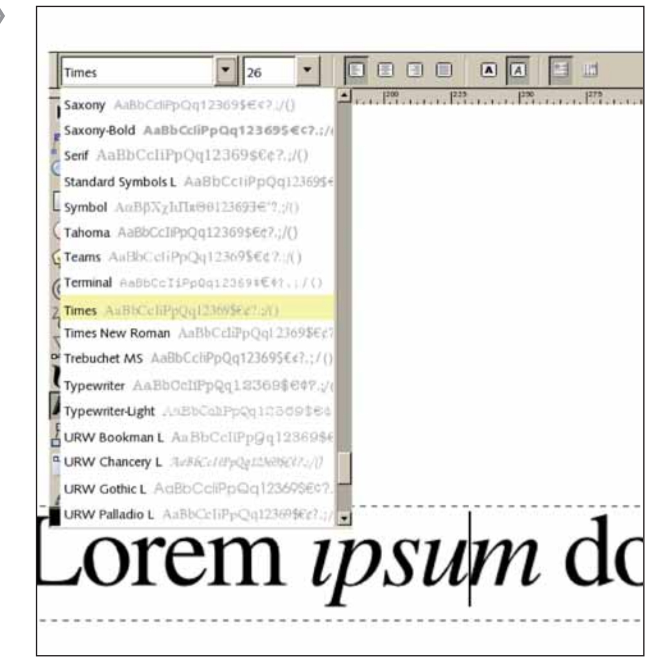
4 Диалог «Слои»

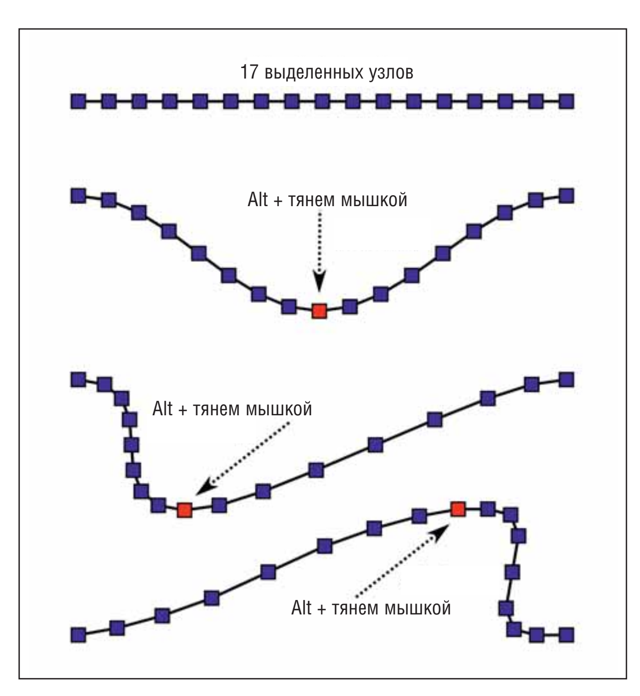
Новое диалоговое окно Слои (Layers), которое открывается по Ctrl+Shift+L, представляет слои документа не в виде списка, а в виде дерева – слои в *Inkscape* могут быть вложены один в другой. Ветви дерева могут разворачиваться и сворачиваться, можно включить/отключить прозрачность каждого слоя или запереть/освободить его. Внизу есть кнопки для добавления слоя или удаления текущего, смены порядка слоев и изменения общей прозрачности каждого слоя. »



Большинство клавиатурных комбинаций можно настроить редактированием несложного XML-файла. Образец файла клавиатурной эмуляции *Xara Xtreme LX* прилагается.

»



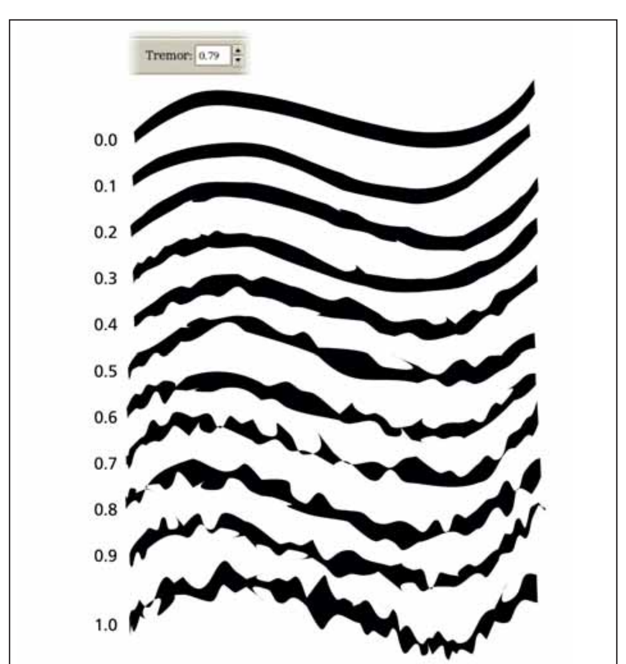


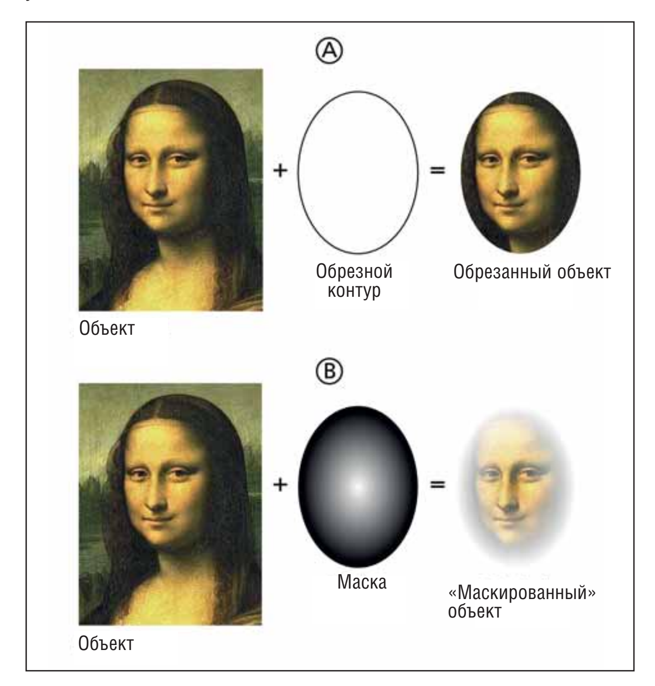
5 Управление текстом

В текстовом инструменте версии 0.43 бросалось в глаза отсутствие приличной панели управления. Единственным средством настройки свойств текстового объекта было диалоговое окно Текст и Шрифт (Text And Font), Ctrl+Shift+T. Сейчас большинство из этих свойств можно настраивать через органы управления в окне редактирования: гарнитуру, размер, полужирный/курсивный варианты, выравнивание, направление письма. Список шрифтов включает графические образцы, а в поле названия работает поиск по мере ввода.

6 Основы работы с узлами

Пластика узловых точек в *Inkscape* явно навеяна подобными функциями программ 3D-моделирования, однако, приложили эти идеи к узлам 2D-контура в векторном редакторе, похоже, впервые. Нужно всего-навсего включить инструмент Узлы (Nodes), выделить несколько точек и перетащить одну из них при нажатой клавише Alt. Выделенная часть линии плавно изогнется, а узлы и их рукоятки сместятся в соответствии с их удаленностью от перетаскиваемого узла.





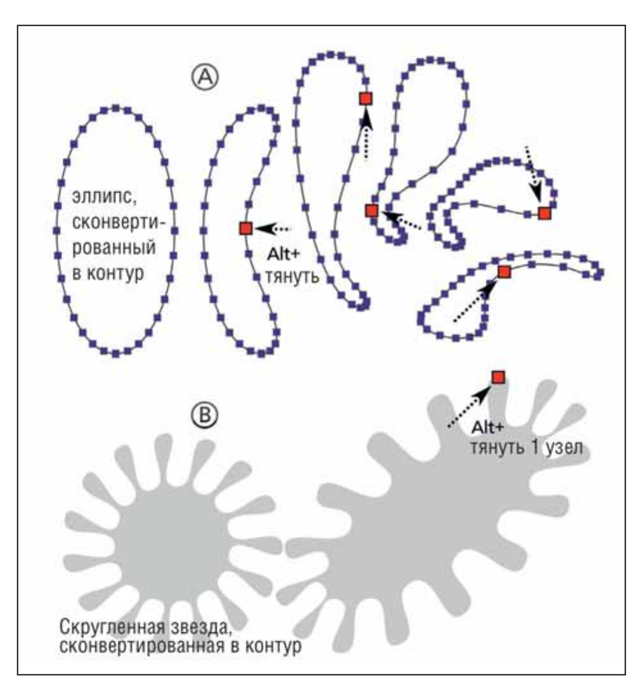
9 Каллиграфический тремор

Как и было предсказано в занятии LXF81, чудесный инструмент Каллиграфическое Перо в *Inkscape 0.44* получил новый настраиваемый параметр: Тремор. При нулевом треморе (по умолчанию) перо производит плавные штрихи постоянной ширины, а с тремором 1.0 штрихи дико искривляются, становятся зазубренными и неопрятными. Очень рекомендую всегда применять тремор, хотя бы легкий: рисунки при этом выглядят более естественно и непринужденно.

10 Вырезки и маски

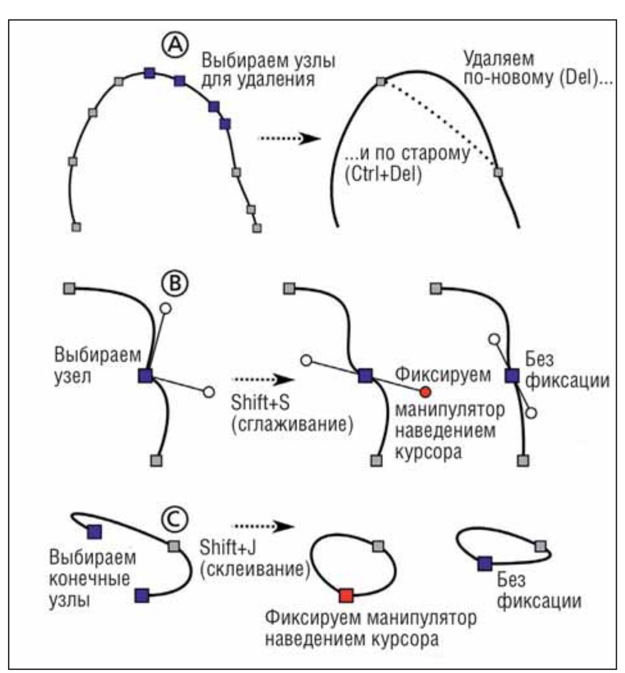
Две важных функции SVG теперь поддерживаются официально: обрезной контур (clipping path) и маски. Обрезной контур, примененный к объекту, ограничивает его видимость областью, которая попала внутрь контура (A). В этом случае стиль контура игнорируется – важна лишь его форма. Маской может быть любой объект, не только контур: белые области маски оставляют цвета и прозрачность объекта, на который накладывается маска, неизменными; под черными же соответствующие места объекта прозрачны (B).

» На диске Версия 0.44 в виде Автопакета.



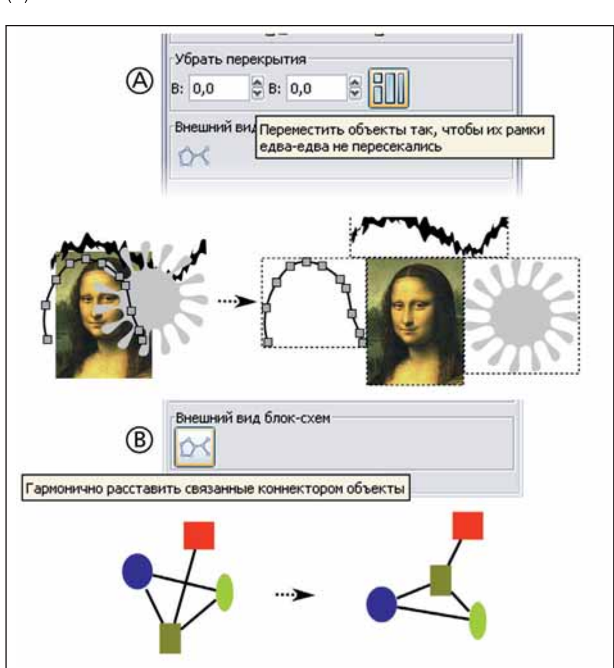
7 Продвинутая обработка узлов

Простота обращения с узлами может переменить приемы вашей работы с кривыми в *Inkscape*. Берете, например, эллипс, преобразуете в контур, выбираете на нем несколько точек нажатиями Ctrl+A и Ins, выделяете часть из них и перетаскиванием при нажатой Alt получаете плавно изогнутую фигуру (A). Вероятно, еще важнее то, что появилась возможность плавно и естественно, без утомительной настройки сотен отдельных точек, изменять сложные контуры (B).



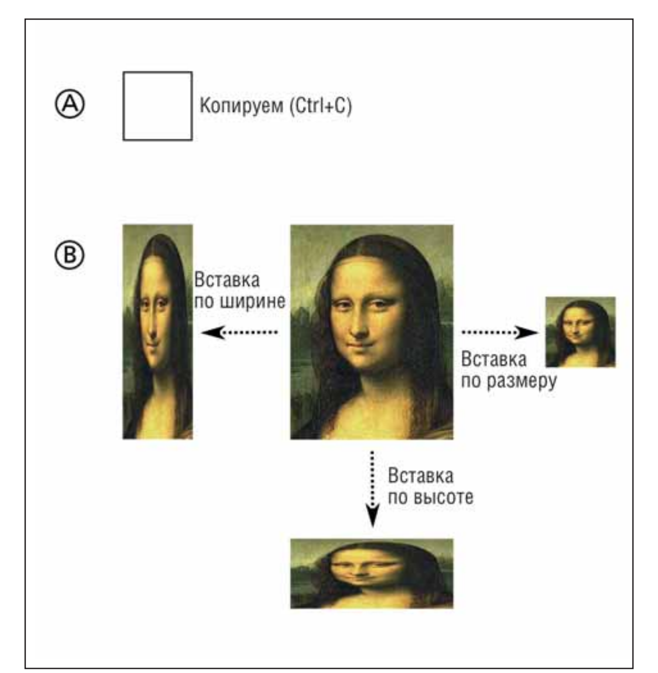
8 Улучшения в редактировании узлов

Если вы удаляете одну или несколько точек в середине кривой, *Inkscape* теперь пытается сохранить форму соответствующего участка. Прежнее поведение при удалении возможно получить при нажатии Ctrl+Del (A). Если вы конвертируете линию с точками в гладкий или симметричный вид (Shift+S или Shift+Y), можно удерживать указатель мыши над одной из рукояток, чтобы зафиксировать ее положение (B). Аналогично, удерживая курсор, можно зафиксировать одну из двух соединяемых точек (Shift+J) (C).



11 Устранение перекрытий

Есть две заметные добавки в диалоговом окне: Выравнивание и распределение (Align and Distribution), Ctrl+Shift+A. Кнопка Убрать перекрытия (Remove Overlaps) независимо от количества выделенных одновременно объектов сдвигает их ровно настолько, чтобы их границы не перекрывали друг друга (A). Еще одно новшество – кнопка Внешний вид блок-схем (Connector Network Layout), которой можно изменить порядок соединений, сделанных инструментом Connector, минимизируя пересечения.





12 Заключительный шаг: Вставка по размеру

В заключение рассмотрим простую, но полезную, функцию: вставку по размеру. Группа команд в подменю меню Правка (Edit) позволяет вогнать выделенный объект (объекты) в размер последнего объекта, скопированного в буфер обмена. Можно вставлять в соответствии с шириной, длиной или обоими измерениями сразу (B), указав, применять ли эту функцию в целом или для каждого выделенного объекта в отдельности. LXF

Скорая помощь

Новый экспортный модуль PDF выводит документ в формате PDF 1.4, который «понимает» прозрачность и градиенты (хотя еще не каждая «гляделка» PDF сможет их отобразить).



Через месяц Я проведу углубленное сравнение *Inkscape* и *Xara Xtreme LX*.



Настоящая безопасность Как воспользоваться умными утилитами Linux и защитить вашу машину.

Безопасность:

ЧАСТЬ 2: Лишние сервисы в вашей машине могут работать на врага. **Д-р Крис Браун** ищет их и отстреливает, отгоняя злоумышленников.



Снова здравствуйте, и добро пожаловать на второй урок по безопасности Linux. В прошлом месяце мы обсуждали важные вещи, касающиеся паролей, и я показал вам, как защититься от некоторых эксплойтов этапа загрузки, дающих привилегии root. Я также объяснил, как использовать *sudo* для контроля над повышением привилегий пользователей, не разглашая администраторский пароль. В этом месяце мы научимся отключать в системе ненужные сервисы, ужесточим права доступа к файлам и познакомимся с некоторыми инструментами, включая *Bastille*, которые помогут нам в этом.

В недалеком прошлом, устанавливая Unix-систему, я обнаруживал, что практически каждый сетевой сервис по умолчанию был запущен. Telnet, FTP и *rlogin*? Да, все включены. NFS и NIS? Работают. Запускались даже такие тривиальные сервисы, как Chargen, ECHO и Daytime. Unix-система как бы говорила: «Смотрите, как много сервисов, разве это не здорово?» Конечно, Интернет в то время был поспокойнее, прямо-таки Хоббитон в киберпространстве. Сейчас это скорее поле битвы, больше похожее на Мордор – в результате в установках Linux-систем по умолчанию не включены никакие сетевые сервисы, кроме *sshd*. Это пример соблюдения важного принципа: не запускайте сервисы, которые вам не нужны.



Наш эксперт

Д-р Крис Браун независимый инструктор по Linux, имеет степень доктора наук по физике элементарных частиц, сертифицированный специалист Novell и Red Hat. Недавно написал книгу о SUSE для издательства O'Reilly.

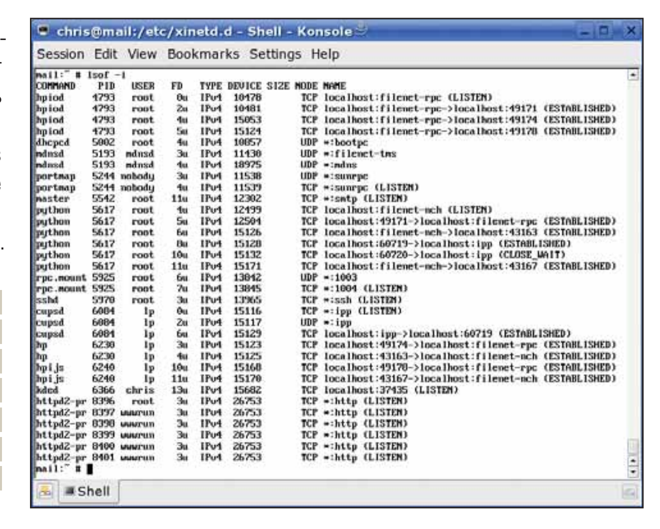
Часть 1: Просмотр запущенных сервисов

Самый непосредственный способ проверить, какие сервисы запущены на вашей машине – запрос открытых TCP и UDP-портов, который можно сделать командой lsof -i. На **рис. 1** показан пример вывода этой команды, выполненной на моем ноутбуке; должен признаться, я не очень в курсе, что именно там понаписано. Ну, portmap – это RPC-демон, необходимый для работы службам NFS и NIS, a master – агент доставки почты postfix, а cupsd – что-то насчет печати. Но зачем здесь какие-то процессы http2 (*Apache*)?

Угадать назначение всех этих таинственных демонов – задача не из легких. Проверьте, есть ли man-страница для сервиса или посмотрите иерархию запущенных процессов. Например, pstree показывает, что hp и hpijs на **рис. 1** – потомки cupsd, и, значит, имеют отношение к печати. Вот соответствующий фрагмент вывода:

```
$ pstree | head
init─┬─acpid
     ├─cron
     ├─cupsd─┬─foomatic-rip───foomatic-rip─┬─foomatic-rip
     |       |                 └─sh───gs─┬─hpijs
     |       |                           └─sh───cat
     |       └─hp
```

Вред от работы ненужных сервисов невелик, однако теоретически их можно использовать как лазейку в вашей системе. Ошибка типа переполнения буфера способна помочь вредителю заставить сервер выполнять произвольный код. Такие атаки довольно часты и, как ни печально, почти так же часто успешны.



› Рис. 1 Проверка открытых портов на моем ноутбуке с помощью *lsof*.

» Месяц назад Мы укрепляли пароли и права доступа с помощью John The Ripper

LINUX FORMAT на DVD
» Bastille 3.0.9

Ограничить в обслуживании!

Например, из-за уязвимости по переполнению буфера в *ProFTPd*-сервере можно было захватить root-привилегии, скачивая особым образом составленные файлы в ASCII-режиме. Уязвимость была найдена в сентябре 2003 года и давно исправлена; однако дыры постоянно обнаруживаются в Skype, *Adobe Reader*, ядре Linux и других компонентах.

Отключив ненужные сервисы, вы значительно снизите вероятность успеха подобных атак. Сообщество Open Source очень быстро устраняет найденные уязвимости, но это вам поможет, только если вы установите исправленную версию. Фактически, абсолютное большинство атак происходит на те компоненты, для которых заплатки уже доступны; тем умнее будет подписаться на рассылки по обновлениям безопасности вашего дистрибутива, например, *Up2date* от Red Hat, и регулярно их применять. Под «регулярно» я понимаю не каждые три месяца (даже три дня могут оказаться роковыми). Атакующие не станут дожидаться вашего очередного ежеквартального обновления.

Скорая помощь

На сервисы вроде Telnet и FTP, некогда рабочие лошадки Интернета, сейчас смотрят косо, поскольку они принимают пароль и имя пользователя по сети в виде открытого текста. Как ни смешно, анонимный FTP – и тот гораздо безопаснее, как раз потому, что не требует для соединения имени пользователя и пароля.

Часть 2: Отключение ненужных сервисов

А как их отключить? В Fedora или Red Hat Linux вы можете остановить сервис *httpd* так:

```
service httpd stop
```

В SUSE эквивалентной командой будет

```
service apache2 stop
```

Это всего лишь остановит сервис «здесь и сейчас», а на самом деле нужно предотвратить его запуск во время загрузки системы. Вы можете сделать это программой *chkconfig*. Например, чтобы отключить автоматический запуск Fedora или Red Hat, выполните следующую команду:

```
chkconfig --del httpd
```

В SUSE такой командой будет:

```
chkconfig --del apache2
```

Chkconfig управляет сложным набором символических ссылок для загрузки сервисов на различных уровнях запуска (runlevel). Команда

```
chkconfig --list
```

покажет вам статус всех сервисов, управляемых подобным образом, а также выведет их уровень запуска.

Отключение ненужных сервисов – это один из способов оградить ваш черный ход от плохих парней. А лучше вообще удалить ненужные сервисы: нет программы – нет проблемы, даже если кто-то все-таки проник в вашу систему. Не нужен вам, например, инструментарий разработчика (компилятор C и т.д.), так и не устанавливайте его: глядишь, руткит в вашей системе и не скомпилируется. Если вы настраиваете сетевой сервер, незачем устанавливать X-сервер и графическое рабочее окружение; и т.д.

Меньше – лучше

На моей тестовой Fedora Core 5, когда я запускаю графический рабочий стол и вхожу в систему, вкалывает уже более 20 дополнительных процессов. Некоторые из них запущены от имени root. Знаю ли я, что они делают? Нет. Знаю ли я, что им можно доверять? Нет. Даже если работа каждого из них в отдельности безопасна, могу ли я быть уверен, что их комбинация не создает побочных эффектов? Нет. И никто не может. Смысл прост: чем меньше сервисов установлено и запущено, тем меньше шансы злоумышленника на обнаружение уязвимости. »

Ограничение на вход в систему от имени root

В прошлом месяце мы обсудили опасности входа в систему от имени суперпользователя и призвали к использованию команды *su* для получения прав root только при необходимости. Вы можете пойти дальше и сделать невозможным вход в систему суперпользователя, оставив единственным способом получения root-привилегий команду *su*. Это легко сделать для программы login: Linux допускает сессии root только в терминалах, перечисленных в **/etc/security**. Если этот файл пуст, зайти как root из терминала будет невозможно. (Будьте осторожны! Если файл удалить, то войти в систему как root вы сможете отовсюду).

Отключение входа под root в KDE и Gnome немножко сложнее. Обычно в конфигурационном файле есть параметр вроде AllowRootLogin или AllowRoot, который должен быть установлен в значение *false*, но найти его нелегко [проще воспользоваться графическими утилитами настройки KDM/GDM, – прим. ред.]. Очевидное преимущество – предотвращение входа напрямую как суперпользователя. Другое преимущество использования *su* – *su* оставляет записи в **/var/log/messages** или **/var/log/secure**, так что вы сможете выяснить, кто что делал от лица root и когда.

Вы также можете ограничить использование *su* определенными группами. Группы wheel и root обычно для этого и используются, однако вы можете определить любую группу:

```
chgrp wheel /bin/su
chmod 4550 /bin/su
```

Комбинирование этих методов приводит к интересной ситуации. Даже если кто-нибудь подслушает пароль в кафетерии, он не сможет получить права суперпользователя, потому что напрямую от root заходить в систему нельзя, и через *su* получить права тоже не удастся, пока пользователь не в нужной группе.

Если вы решите применять эти приемы, убедитесь, что добавили хотя бы одного пользователя в группу wheel. Иначе вы сами не сможете получить административный доступ и вам придется загружаться с какого-нибудь LiveCD для исправления ситуации.

Часть 3 Системные инструменты ограничения прав

» Есть несколько утилит, специально спроектированных для укрепления безопасности вашей системы (как правило, для отключения ненужных компонентов и «ужесточения» прав доступа к файлам) и предлагающих более систематический подход, нежели чисто «ручной». *Bastille* – одна из лучших подобных программ. *Bastille* не только укрепляет защиту, она еще и поможет разобраться, как работают системы безопасности. На время написания статьи последняя версия (3.0.9) была доступна по адресу http://prdownloads.sourceforge.net/Bastille-linux/Bastille-3.0.9-1.0.noarch.rpm (ее можно взять с нашего диска).

Bastille написана на Perl, так что вам потребуется интерпретатор этого языка и библиотеки Perl-Tk или Perl-Curses для пользовательского интерфейса (возможно, в вашем дистрибутиве эти пакеты устанавливаются по умолчанию). После установки просто запустите **Bastille** с опцией **-x** для интерактивной графической сессии или с опцией **-a** для оценки безопасности вашей системы.

Интерактивная сессия учинит вам допрос о ваших опциях по безопасности. На **рис. 2** показан один разделов этого допроса – все разделы сгруппированы в 15 категорий, перечисленных слева. Вас спросят, *«Хотите ли вы отключить r*-утилиты?», «Хотите ли вы ограничить срок жизни паролей?»* и многое другое. При первом запуске вы, возможно, затруднитесь ответить на некоторые вопросы. В этом случае лучшим выходом будет сделать себе чашечку кофе, возможно, устроить срочное совещание, и все-таки решить, чего вы хотите. Это часть процесса определения политики безопасности, и это важный шаг по защите вашей машины. Кстати, *Bastille* не сделает ничего до тех пор, пока вы не ответите на все вопросы и не подтвердите изменения.

Также вы можете запустить *Bastille* в оценочном режиме (**опция -a**): никаких изменений она делать не будет, лишь выведет отчет в виде HTML-страницы и покажет его в браузере. Вы можете видеть пример такого отчета на **рис. 3**. Каждая позиция в отчете также является ссылкой на более подробное объяснение.

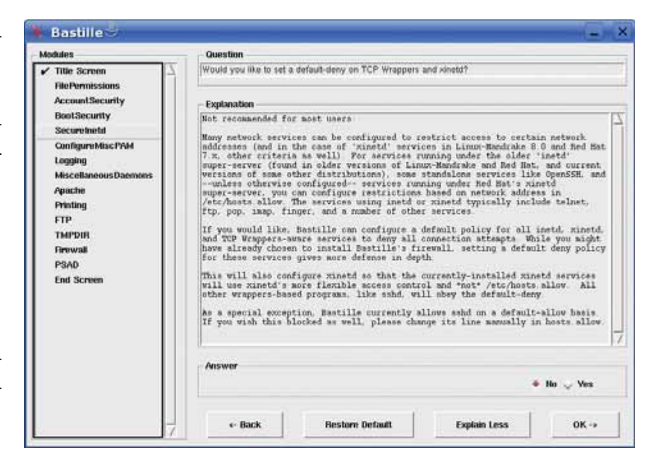
Мастер безопасности SUSE

Теперь давайте рассмотрим мастер безопасности, встроенный в SUSE Linux. Этот компонент, входящий в *YaST*, тоже предназначен для укрепления системы. Он дает возможность привести в порядок множество опций и настраивает соответствующие файлы конфигурации. Для его запуска выберите пункт Локальная безопасность (Local Security) из секции Безопасность и пользователи (Security and Users) конфигуратора *YaST*. На первом экране мастер спросит вас, не хотите ли вы применить один из трех готовых шаблонов безопасности. В порядке увеличения безопасности они выглядят так: домашний компьютер (Home Workstation), сетевая рабочая станция (Networked Workstation) и сетевой сервер (Network Server). Если хотите, можете просто выбрать один из этих профилей и нажать Finish. Это, конечно, довольно тупой способ настройки безопасности системы, но уж точно лучше, чем ничего.

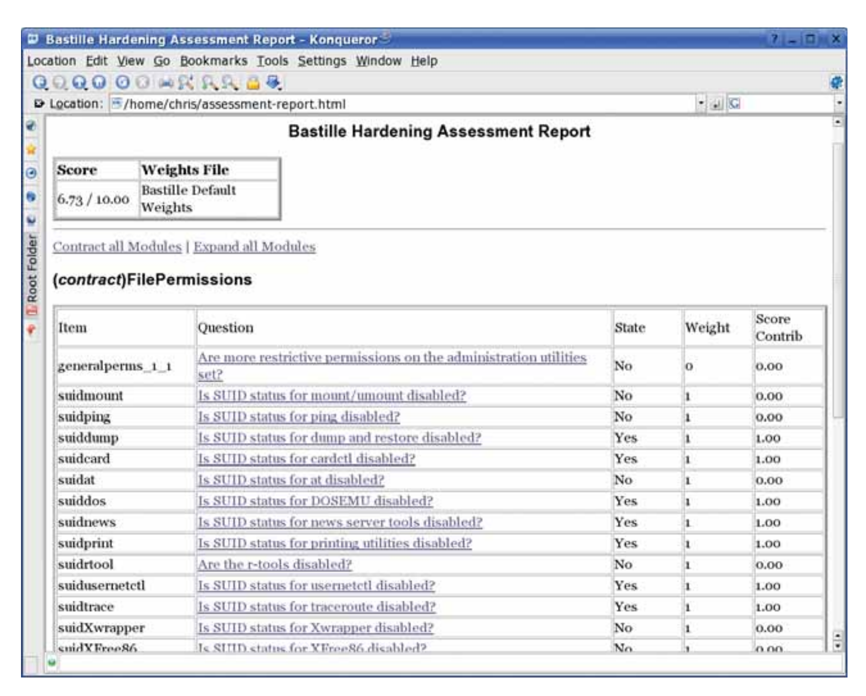
Для более тонкой настройки выберите Custom Settings и нажмите Next. Это перенесет вас в первый из пяти шагов, где вы можете установить различные настройки (однако, как и в случае с *Bastille*, вы должны знать, чего вы хотите). Вы можете включить и отключить проверку качества паролей, о которой мы говорили в первой части в LXF84 (вы ведь читали первую часть?). Можете выбрать метод шифрования пароля и другие параметры. Можете контролировать задержку после неудачной попытки входа в систему (увеличение этой задержки, несомненно, замедлит ручной подбор пароля, но таким способом пытаются проникнуть в систему разве что самые простодушные злодеи). Вы можете настроить действие системы по нажатию Ctrl+Alt+Del и указать, какие пользователи могут завершать работу системы.

Также возможно проконтролировать, какие **.** (текущие каталоги) должны быть включены в пути поиска для супер- и других пользователей. Зачем это надо? Допустим, вы зашли в оболочку и запускаете какую-нибудь обычную программу, вроде *cp*. Какая программа запустится? Ответ такой: любой файл с именем *cp*, который оболочка первым найдет в путях поиска. Вы надеетесь, что это cp, расположенная в директории **/bin**. Однако если включить **.** (текущий каталог) в путь поиска (особенно перед **/bin**), есть реальная опасность, что запускаемая команда *cp* – одна из тех, что злоумышленник подкинул вам в текущую директорию. Таким способом можно запускать любые программы с вашими правами. Если вы суперпользователь, последствия могут быть особенно серьезными. По этой причине разумно не включать **.** в пути поиска. Захочется запустить какую-нибудь программу, находящуюся в текущей директории – явно введите **./myprog** вместо **myprog**.

На последнем шаге вы можете выбрать, как жестко система будет обращаться с правами доступа к файлам. Тут есть три варианта: простой, безопасный и параноидальный. На самом деле этот экран – всего лишь интерфейс к программе *chkstat*, которая проверяет (и при необ-



› **Рис. 2 Один из многих разделов вопросника *Bastille*. Да, он сероват, но безопасность не в красоте.**



Bastille Hardening Assessment Report

Score	Weights File
6.73 / 10.00	Bastille Default Weights

Contract all Modules | Expand all Modules

(contract)FilePermissions

Item	Question	State	Weight	Score Contrib
generalperms_1_1	Are more restrictive permissions on the administration utilities set?	No	0	0.00
suidmount	Is SUID status for mount/umount disabled?	No	1	0.00
suidping	Is SUID status for ping disabled?	No	1	0.00
suiddump	Is SUID status for dump and restore disabled?	Yes	1	1.00
suidcard	Is SUID status for cardctl disabled?	Yes	1	1.00
suidat	Is SUID status for at disabled?	No	1	0.00
suiddos	Is SUID status for DOSEMU disabled?	Yes	1	1.00
suidnews	Is SUID status for news server tools disabled?	Yes	1	1.00
suidprint	Is SUID status for printing utilities disabled?	Yes	1	1.00
suidrtool	Are the r-tools disabled?	No	1	0.00
suidusernetctl	Is SUID status for usernetctl disabled?	Yes	1	1.00
suidtrace	Is SUID status for traceroute disabled?	Yes	1	1.00
suidXwrapper	Is SUID status for Xwrapper disabled?	No	1	0.00
suidXFree86	Is SUID status for XFree86 disabled?	No	[illegible]	0.00

› **(Рис. 3) Главное назначение отчетов *Bastille* – затруднить атаки на вашу систему, а заодно это неплохой инструмент обучения администраторов.**

ходимости подправляет) права доступа и владельца ключевых системных файлов. Программа читает один или более permission-файл в директории **/etc** (*см. врезку внизу справа*). Синтаксис каждой строки в этих файлах такой:

```
имя файла   владелец:группа   режим
```

Например, вот несколько строк из файла **/etc/permissions.secure**:

```
/usr/bin/passwd        root:shadow     4755
/usr/bin/crontab       root:trusted    4750
/etc/fstab             root:root       644
/var/spool/fax/archive fax:uucp        700
```

Права доступа (**режим**) указываются в восьмеричном виде. Восьмеричное значение 4000 устанавливает SUID-бит (*см. врезку вверху справа*), другие три цифры отвечают за права чтения, записи и выполнения для владельца, группы и всех остальных.

Для запуска *chkstat* с настройками из файла **permissions.secure** (например), просто укажите имя файла в качестве аргумента. На моей тестовой системе команда выглядит так:

```
# chkstat /etc/permissions.secure
```

В моем случае на экран ничего не выводилось. Однако запустив *chkstat* с настройками из **permissions.paranoid**, вы увидите массу предупреждений, в основном о файлах конфигурации, доступных несуперпользователям, и о программах с установленным SUID-битом.

Запуск *chkconfig* с опцией **--set** велит программе исправлять любые проблемы автоматически. Давайте нарочно сделаем права доступа на один из системных файлов слишком слабыми и перезапустим программу:

```
1 # chmod 666 /etc/fstab
2 # chkstat /etc/permissions.secure
3 Checking permissions and ownerships – using the permissions files
  /etc/permissions.secure
4 /etc/fstab should be root:root 0644. (wrong permissions 0666)
5 # chkstat --set /etc/permissions.secure
6 Checking permissions and ownerships – using the permissions files
  /etc/permissions.secure
7 setting /etc/fstab to root:root 0644. (wrong permissions 0666)
8 # ls -l /etc/fstab
9 -rw-r--r-- 1 root root 670 Oct 11 09:35 /etc/fstab
```

В первой строке я сменил права доступа к файлу **/etc/fstab**. В строках 2–4 я запустил *chkstat*, и он сообщил о проблеме. В строках 5–7 я снова запустил *chkstat*, с опцией **--set**, и в строках 8–9 убедился, что chkstat корректно решил проблему.

Давайте свяжем это с настройками привилегий из *YaST*. *YaST* устанавливает в **/etc/sysconfig/security** строку вроде этой (вариант для профиля 'easy'):

```
PERMISSION_SECURITY="easy local"
```

Строка велит *chkstat* запускаться с файлами **permissions.easy** и **permissions.local**. Программа *chkstat* запускается Suseconfig, скриптом, который *YaST* использует для применения проделанных изменений. Это значит, что права доступа к файлам будут проверяться каждый раз, когда вы измените что-либо в *YaST*.

Есть и другие утилиты для подобных целей. Я планировал рассказать о Titan, который особенно популярен в Solaris, но он выглядит бледно после выпуска *Sun Security Toolkit от JASS*. Есть еще Sussen, сканер уязвимостей, основанный на OVAL ((Open Vulnerability and Assessment Language). Это один из новоиспеченных C#/Mono-проектов, о котором я надеюсь рассказать на следующих уроках. На сегодня все; храните безопасность до следующего месяца! LXF

SUID-программы

В основе увеличения привилегий в Linux стоят SUID-программы. Идея состоит в запуске программы с правами владельца исполняемого файла, а не того, кто производит запуск. Это поведение зависит от SUID-бита, который является частью прав доступа к файлу.

Один из лучших примеров такой программы – **/usr/bin/passwd**. SUID-бит можно обнаружить в выводе ls -l:

```
ls -l /usr/bin/passwd
-r-s--x--x 1 root root 21944 Feb12 08:15 /usr/bin/passwd
```

В позиции четвертого символе вместо x стоит s. Обратите внимание, что файлом владеет root. Поэтому программа может записывать новый хэш в shadow-файл, прав на запись [и чтение, – прим. ред.] в который у обычного пользователя быть, конечно же, не должно. Программа *sudo*, рассмотренная месяц назад, также является SUID-программой – чтобы позволить запускать программы под другими идентификаторами. (Если программа запускается с привилегиями root, она может сменить текущий ID на любой.)

SUID-программы, особенно принадлежащие root – всегда возможная уязвимость. Можно ли верить, что такая программа делает только то, что нужно? А вдруг она приписывает данные к файлам? Как грамотный системный администратор, будьте бдительны по поводу любых SUID-программ. Их можно найти так:

```
find / -perm +4000 -user root
```

Установить SUID-бит можно таким образом:

```
chmod u+s filename
```

а удалить

```
chmod u-s filename
```

Вся идея SUID-бита запатентована дедушкой Unix, Деннисом Ричи [Dennis Ritchie]. Номер патента США – 4135240; он был выдан 16 января 1979 года.

Права доступа к файлам в SUSE

Права доступа	Уровень безопасности
Permissions	Простейший набор прав. Более мягкие права доступа, нежели те, что указаны в этом файле, не просто небезопасны, а уже надпись на могиле!
Permissions.easy	Довольно мягкие права для использования в настольных системах с одним пользователем.
Permissions.secure	Более безопасный набор прав для многопользовательских или сетевых систем. Вот комментарий из файла: «Главная задача этих прав – заставить простые вещи, вроде смены паролей или использования сетевых программ, правильно работать для непривилегированного пользователя».
Permissions.paranoid	Очень безопасный набор прав. Похож на предыдущий вариант, однако SUID и SGID-биты сбрасываются. Эти настройки подходят для сетевых сервисов или межсетевых экранов, в которые обычным пользователям делать нечего.
Permissions.local	Это специальный файл для добавления прав на дополнительные программы. По умолчанию он пуст.

» **Через месяц** Мы рассмотрим сканеры уязвимостей – Nmap и Nessus.

Ogre: Добавим вражьих

ЧАСТЬ 4: Наскучило бродить одному по бескрайним просторам? **Пол Хадсон** подскажет вам, как добавить плохих ребят...





Наш эксперт

Пол Хадсон написал три книги по Linux и одну по PHP, он участвует в двух GPL-проектах для Mono на SourceForge. Пол любит Emacs.

Солнце встает, освещая старый бревенчатый домик, а вода плещется у краев покатого ландшафта – наступает новый день в игре *Висельник Чед*. За последние три урока мы создали нашему игроку живописный остров, где он может порезвиться. Логичный следующий шаг – населить остров роботами-убийцами. Как, вы в этом не уверены? Тогда пришлите нам запрос на создание учебника по игре «Сажаем цветочки», и мы посмотрим, что можно сделать, а на данном уроке займемся роботами. Сотней здоровенных, вооруженных до зубов, злобных роботов. Да-с.

Наши роботы будут первым шажком на пути к искусственному интеллекту (ИИ), именно здесь и начинается настоящая потеха: неважно, как *Ogre* работает с 3D-графикой, сам по себе он нечеловечески скучен. Все, что выходит за рамки влияния *Ogre*, придется писать самим. Поверьте, это забавно!

На сей раз мы создадим новый класс, отслеживающий перемещение роботов. Я не большой любитель объектно-ориентированного программирования – обычно мои классы превращаются в штуки типа struct; и я, скорее всего, не один такой. Кому не по душе моя манера делать все переменные публичными, пишите письма на имя **/dev/null**.

В процессе создания роботов вы познакомитесь с кватернионами (которые, наряду с 'shazbat', являются моим любимым словом), анимацией и Стандартной библиотекой шаблонов C++ (*Standard Template Library* или *STL*). Если вы раньше не использовали *STL*, вам вполне простительно решить, что это плод труда двух комитетов из разных стран, говорящих на разных языках – но, отстранившись от мелочей, вы поймете, как все просто.

Наши враги будут управляться специальным классом, который я назову CChadEnemy, так что создайте файлы **chademeny.h** и **chadenemy.cpp** (да позаботьтесь о включении файла **chadenemy.cpp** в **Makefile**). Нам требуется, чтобы у робота была своя сущность (каркас робота) и узел сцены, чтобы мы могли помещать его в сцену. Вспомните, как мы передвигали игрока (LXF83): используя луч для нахождения высоты игрока над ландшафтом. Надо придумать что-то подобное для каждого из наших врагов, поэтому вместо указания позиций в **chad.cpp** применим метод SetPos().

Вот файл **chad.h**:

```
#include "Ogre.h"
#include "OgreFrameListener.h"
#include "OgreEventListeners.h"
#include "OgreKeyEvent.h"
using namespace Ogre;
class CChadEnemy {
        public:
            SceneManager* m_SceneMgr;
            Entity* m_Entity;
            SceneNode* m_Node;
            int m_Speed;
            CChadEnemy(SceneManager* scenemgr);
            void SetPos(Vector3 dest);
};
```

Каждый враг создается своим конструктором, CChadEnemy(). В нем будет происходить загрузка каркаса, создание узла сцены, затем добавление в менеджер сцены (вы заметили указатель на менеджер сцены в **chad.h?**). Каркас робота предоставляется *Ogre* и должен находится в вашем каталоге с медиа-ресурсами, но мы уменьшим размер робота вчетверо, вот этот код:

```
CChadEnemy::CChadEnemy(SceneManager* scenemgr) {
            m_SceneMgr = scenemgr;
            static int EnemyNum = 0;
            char enemyname[32];
            sprintf (enemyname, "Robot %d", ++EnemyNum);
            m_Entity = m_SceneMgr->createEntity(enemyname,
            "robot.mesh");
            m_Node = m_SceneMgr->getRootSceneNode()
            ->createChildSceneNode();
            m_Node->attachObject(m_Entity);
            m_Node->scale(0.25,0.25,0.25);
            m_Speed = 5;
}
```

Строки с EnemyNum, enemyname и sprintf() раздражают своим уродством, но для *Ogre* это в порядке вещей. Проблема здесь в том, что каждой загружаемой сущности требуется дать уникальное имя, а достичь этого можно, отслеживая, сколько врагов уже создано с помощью переменной EnemyNum, и добавляя это значение к слову Robot. EnemyNum объявлена как статическая переменная, то есть ее разделяют все экземпляры класса.

Месяц назад Мы создали дом, эффект смены дня и ночи, а также немного теней.

роботов-умников

Я не собираюсь печатать метод SetPos(), потому что по большей части он идентичен методу SetPos() для игрока; вы можете просто его скачать из исходного кода с web-сайта Linux Format (*см. сноску на стр. 88*).

Стандартная библиотека шаблонов

Теперь у нас хватает кода, чтобы населить остров врагами-роботами, но сначала надо придумать, как их хранить. Метод должен быть гибким: незачем объявлять переменную для каждого робота или заводить статические массивы, потому что желательно создавать роботов по запросу. Решение состоит в использовании *STL*: это коллекция стандартных абстрактных типов данных для C++. Пусть слово «абстрактные» вас не пугает: *STL* чрезвычайно полезна, а ее типы данных обычно легко представить.

Для управления врагами нам потребуется тип данных «вектор» – массив некоторых величин, в котором нас интересует только их значение. Добавление «вражеского» вектора потребует сделать два изменения в **chad.h**. Вставьте следующие две строчки после директив #include:

```
#include "chadenemy.h"
#include <vector>
```

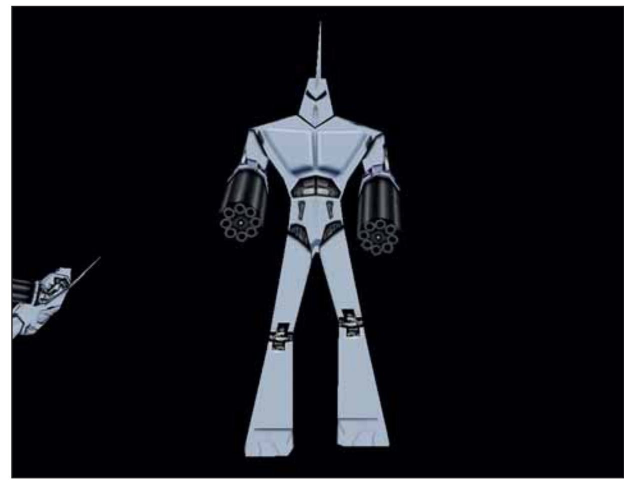
Теперь вставьте эту строку в конце определения класса CChadGame:

```
std::vector<CChadEnemy*> Enemies;
```

Далее припишите к методу createOutdoorScene() следующий код:

```
for (int i = 0; i < 100; ++i) {
CChadEnemy* enemy = new CChadEnemy(m_SceneMgr);
enemy->SetPos(Vector3(rand() % 1000,0, rand() % 1000));
Enemies.push_back(enemy);
}
```

Этот код создаст 100 новых врагов, задаст им произвольные позиции на карте (Y установлен равным 0: метод SetPos() потом переопределит это значение с учетом высоты местности), затем добавляет врагов в вектор Enemies. Если вашей системе больше трех лет от роду, врагов лучше взять поменьше.



› **Ogre поставляется со всеми мыслимыми моделями – от греческих статуй до ниндзя, но у этого типа самые классные пушки!**

Вот и все, что потребовалось для добавления роботов – теперь скомпилируйте игру и запустите ее!

Двигаемся дальше

Наши роботы стоят столбом, напоминая батарейки из рекламы Duracell, которые демонстрируют, что другие уже рухнули. Попробуем их оживить: пусть выберут какой-то пункт назначения на карте и отправятся туда, обуреваемые злыми намерениями. Для этого нам потребуется внести следующие изменения:

1. **Назначить каждому роботу точку назначения и направление.**
2. **Сказать каждому роботу, сколько времени прошло с его последнего перемещения (поэтому они двигаются с постоянной скоростью, независимо от скорости вашего компьютера).**
3. **Велеть роботам перемещаться при начале каждого кадра.**

Мы можем разрешить первые два пункта, отредактировав файл **chadenemy.h**. Добавьте две строки после определения m_Speed: »

Скорая помощь

Мы привыкли перемещать камеру, но в этот раз я прикрепил ее к собственному узлу сцены и передвигал его. Такой ход позволит в будущем проделывать интересные вещи, например, вызывать метод lookAt() для других узлов, чтобы они взглянули в камеру.

Кватернионы и вы

Рысканье, тангаж и крен (РТК) – известные как угловое вращение Эйлера – так люди представляют себе перемещение в пространстве. Если у вас есть игрушечный самолетик и пара шампуров, то вот как понять эти характеристики:

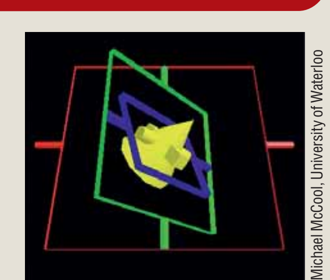
» **Рысканье** – глядя на самолет сверху вниз, проткните его через крышу и пол: он сможет вращаться только влево и вправо.

» **Тангаж** – глядя на самолет сбоку, проткните его с одного борта до другого: он сможет вращаться только вверх и вниз.

» **Крен** – глядя на самолет спереди, проткните его от носа до хвоста: он сможет вращаться только вправо или влево; учтите, это не то вращение, что при рысканье!

Проблема традиционного РТК-метода работы с 3D-пространством (кроме того, что жалко дырявить самолетик) – здесь иногда случается складывание плоскостей. Такое происходит, когда какой-либо поворот кратен 90 градусам: в результате на вид две степени свободы превращаются в одну. Вам будет легче осознать, в чем дело, если вы взглянете на рисунок справа: внешняя рамка представляет тангаж, средняя – крен, а внутренняя – рысканье. Если средняя рамка провернется еще влево, то есть попадет в одну плоскость с внешней, то внутренняя рамка будет функционально аналогична внешней рамке, и нам покажется, что степень свободы потеряна.

Кватернионы решают эту проблему, определяя вектор (значения X,Y,Z) плюс угол поворота вокруг этого вектора. Поэтому вместо того, чтобы протыкать наш самолет в трех местах, вращая по отдельности, мы протыкаем его произвольным образом. Такой способ позволяет избегать слипания рамок и проводить гладкую интерполяцию между точками – идеально при полетах с кинокамерой.



› **При приближении зеленой рамки к горизонтальной плоскости синяя рамка начинает вести себя как красная, и мы теряем степень свободы.**



› На прошлом уроке мы поработали неплохо: наши орды злодеев отбрасывают при движении реалистичные тени.

Скорая помощь

Если вы обнаружили, что анимация не работает, проверьте, вызывается ли метод addTime() для каждого робота при вызове его метода Update(). Без этого анимация обновляться не будет: будет проигрываться один и тот же кадр.

```
Vector3 m_Direction;
Vector3 m_Destination;
```

Также вам понадобится дописать определение следующего метода в определении класса:

```
void Update(Real time);
```

Мы будем вызывать метод Update() для каждого робота каждый раз, как только начнется кадр, поэтому робот сможет перемещать сам себя. Временной параметр сообщает роботам, сколько времени прошло с момента отрисовки последнего кадра. *Ogre* об этом позаботился: взгляните на метод CChadGame::frameStarted() в файле **chad.cpp**, и увидите следующее:

```
bool CChadGame::frameStarted(const FrameEvent& evt) {
```

Параметр FrameEvent сообщает множество характеристик текущего кадра, включая время, прошедшее между кадрами. Проблема состоит в том, что мы затем вызываем frameStartedOutside() или frameStartredInside() и не передаем этот параметр. Поэтому откройте файл **chad.h** и измените их, чтобы они соответствовали прототипам:

```
bool frameStartedOutside(const FrameEvent& evt);
bool frameStartedInside(const FrameEvent& evt);
```

Вам также понадобится отредактировать файл **chad.cpp**. Метод frameStarted() необходимо подправить, чтобы он передавал параметр требуемому методу, например:

```
bool CChadGame::frameStarted(const FrameEvent& evt) {
        switch (scenemanager) {
                case ST_EXTERIOR_CLOSE:
                        return frameStartedOutside(evt);
                        break;
                case ST_INTERIOR:
                        return frameStartedInside(evt);
                        break;
        }
        return true;
}
```

Каждый раз, когда кадр начинается вне дома, необходимо обновить всех наших роботов с помощью метода Update(). В части кода, получается, надо пройти в цикле по вектору Enemies() и передать данные о кадре, предоставляемые *Ogre*. Поэтому в конце метода frameStartedOutside() (перед строкой 'return line') добавьте небольшой цикл:

```
for (std::vector<CChadEnemy*>::iterator iter = Enemies.begin();
iter != Enemies.end(); ++iter) {
  CChadEnemy* enemy = (*iter);
  enemy->Update(evt.timeSinceLastFrame);
}
```

Здесь *STL* применяется в большей степени, и наш код напоминает синтаксическую окрошку. На самом деле этот цикл проходит от начала до конца вектора, используя *STL*-итератор. Можно, конечно, получить прямой доступ к элементам вектора через индексы, но итераторы более предпочтительны. Для каждого врага в нашем векторе мы вызываем метод Update() и посылаем отрезок времени, прошедший с прошлого кадра. Именно тут наши роботы должны проделать всё, что им полагается – сейчас мы просто заставим их двигаться.

Пора написать сам метод Update(). Вставьте следующий код в конец файла **chadenemy.cpp**:

```
void CChadEnemy::Update(Real time) {
   if (m_Destination == Vector3::ZERO) {
      m_Destination = Vector3(rand() % 1000,0, rand() % 1000);
      m_Direction = m_Destination - m_Node->getPosition();
   } else {
      Vector3 ourpos = m_Node->getPosition();
      ourpos.y = 0.0f;
      m_Direction = m_Destination - ourpos;
       Real dist = m_Direction.normalise();
      Real movespeed = m_Speed * time;
      if (dist < movespeed) {
         m_Destination = Vector3::ZERO;
      } else {
         // необходимо переместиться ближе к точке назначения
         Vector3 newpos = m_Direction * movespeed;
         SetPos(newpos);
      }
   }
}
```

Код включает две ветви: если врагу уже задано направление движения, то он по нему и движется, если же нет, направление выбирается произвольно. Ветвь определяется в самом начале: если вектор m_Destination равен Vector::ZERO (по умолчанию, пустой вектор), то необходим выбор точки назначения. Зная эту точку, мы можем вычислить направление простым вычитанием координат текущей позиции (все векторные операции *Ogre* берет на себя).

Пункт назначения: Чед

Если роботу задан путь, то мы берем его текущую позицию и точку назначения, затем пересчитываем направление. Это не обязательно, потому что направление использует числа с плавающей запятой: достаточно посчитать его один раз и забыть о нем. Но и вреда в этом нет: робот будет корректировать свою позицию при каждом кадре, и придет точно в заданный пункт независимо от расстояния до него.

Мы можем рассчитать длину вектора, вызвав метод normalise(), который вернет нормализованный вектор – имеющий то же направление, но длина его равна 1. Штука хорошая, но на самом-то деле нас интересует значение длины исходного вектора, которое также возвращается методом normalise и в нашем коде является расстоянием от робота до точки назначения. Сохраним это значение в переменной dest.

Теперь – важный момент: если расстояние до точки назначения короче, чем расстояние, на которое собирается переместиться робот (его скорость, помноженная на время, прошедшее с прошлого кадра), значит, он подобрался очень близко, и нам необходимо выбрать новое направление. Это можно сделать, установив вектор m_Destination равным Vector3::Zero, тогда при следующем вызове метода робот случайным образом выберет другую точку назначения.

Код этого урока можно найти на: www.linuxformat.co.uk/mag/hangingchad_lxf85.tar.gz

Если расстояние до пункта назначения больше, чем величина перемещения робота, помножим направление на скорость робота и передвинем его на следующую позицию. Вектор направления уже был нормализован, поэтому все будет работать.

Вот и все – запустите новый код и наслаждайтесь реалистичностью перемещений ваших роботов по ландшафту!

Идем верным путем

Ну, пожалуй, «реалистичность» – слишком сильно сказано: на самом деле роботы просто скользят, по причине отсутствия кода для анимации. Да и вообще, нет ничего, обуславливающего должное направление их перемещения. Давайте это исправим!

В файле **chadenemy.h** добавьте две строки, перед вектором m_Direction:

```
AnimationState* m_AnimationState;
bool WatchingPlayer;
```

AnimationState представляет собой тип данных *Ogre*, ответственный за анимацию сущности: проигрывать ли ее бесконечно, и так далее. Мы хотим, чтобы наши роботы по умолчанию использовали анимацию 'Idle', поэтому скопируйте следующий код в конструктор класса CChadEnemy:

```
m_AnimationState = m_Entity->getAnimationState("Idle");
m_AnimationState->setLoop(true);
m_AnimationState->setEnabled(true);
WatchingPlayer = true;
```

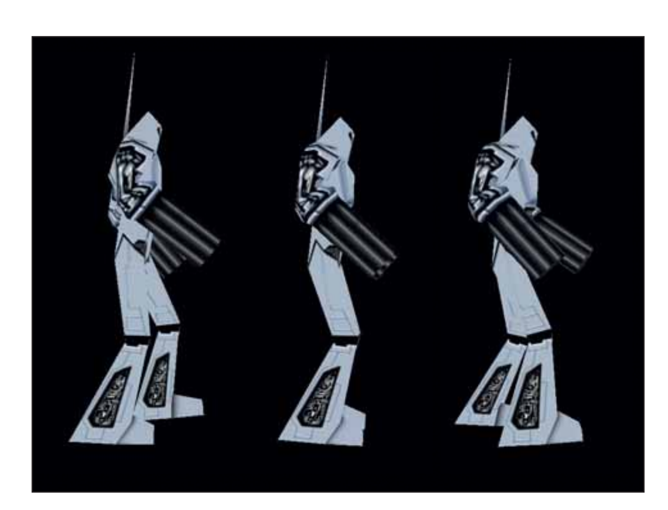
Переменная WatchingPlayer будет первым кусочком искусственного интеллекта: если робот приблизится к игроку на определенное расстояние, пусть остановится и посмотрит на игрока. По умолчанию такая опция установлена для каждого робота; вы скоро узнаете, зачем.

Осталось изменить метод Update(). Первое, что необходимо сделать – обновить анимацию робота, этим займется метод addTime() нашей переменной m_AnimationState(). Он принимает в качестве параметра время, прошедшее с прошлого кадра, которое мы уже знаем, поэтому поместите следующую строку в начале метода Update():

```
m_AnimationState->addTime(time);
```

Далее нам необходимо поместить код, вычисляющий расстояние между текущим роботом и игроком и заставляющий робота предпринять соответствующее действие, если поблизости находится игрок: бросить вышеупомянутый злобный взгляд. Добавьте следующий код (после проверки Vector3::ZERO) внутри другого блока if:

```
Vector3 campos = m_SceneMgr->getSceneNode("camnode")
->getPosition();
Vector3 ourpos = m_Node->getPosition();
Vector3 playerdist = ourpos - campos;
if (playerdist.normalise() < 100) {
   if (!WatchingPlayer) {
      m_AnimationState = m_Entity->getAnimationState("Idle");
      m_AnimationState->setLoop(true);
      m_AnimationState->setEnabled(true);
      WatchingPlayer = true;
   }
   Vector3 orient = m_Node->getOrientation() * Vector3::UNIT_X;
   Vector3 dir = campos - m_Node->getPosition();
   dir.y = 0;
   Ogre::Quaternion quat = orient.getRotationTo(dir);
   m_Node->rotate(quat);
} else {
   if (m_Destination == Vector3::ZERO) {
      // здесь почти ничего не изменилось
      SetPos(newpos);
   }
   // а это новый код
   if (WatchingPlayer) {
      m_AnimationState = m_Entity->getAnimationState("Walk");
      m_AnimationState->setLoop(true);
      m_AnimationState->setEnabled(true);
      WatchingPlayer = false;
   }
}
```



› **Здесь всего три кадра, но можно видеть, что наш враг размахивает руками в такт шагам, нечувствительный ко всему на свете.**

Итак, если робот невдалеке от игрока, мы заставляем его взглянуть, если только он уже не глядит (чтобы избежать лишних вызовов функции). Задача заставить робота смотреть на игрока представляет собой лишь вопрос получения ориентации робота, нахождению направляющего вектора между нами и роботом, а затем преобразованию его в кватернион, используя метод getRotationTo() вектора. Это позволит нам развернуть робота лицом к игроку, передав кватернион в метод rotate().

Если робот находится далеко, выполним проверку с вектором Vector3::ZERO, за которым следует много кода, заканчивающегося вызовом SetPos(). Тут все осталось без изменений, но после этого нам надо проверить, смотрел ли робот на предыдущем шаге на игрока, и если да, то заставить робота снова шагать. Теперь вы понимаете, почему все роботы по умолчанию видят игрока: первый раз при проходе по этому методу, его позиция будет установлена правильно, а анимация работать.

Лимит на код в этом уроке исчерпан. Прокрутите свою игру: мы добавили врагов, передвижение, анимацию и простенький искусственный интеллект – не так уж плохо для одного урока! LXF

Домашнее задание

» Назначьте каждому роботу персональную скорость.

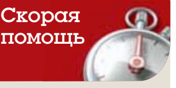
» Ваши роботы идут в некотором направлении и смотрят на игрока, когда он в зоне досягаемости. Теперь заставьте их ходить за игроком по пятам!

» Более опытные программисты могут попробовать выставить направление следующего робота по предыдущему, а первый робот выбирает произвольное направление.

Наше спасибо...

Карта Nvidia GeForce 7800, использованная при разработке этой серии уроков, была предоставлена MSI – спасибо, ребята!

Скорая помощь



У нашего робота имеются и другие анимации, например, 'Shoot' [Стрелять] и 'Die' [Умирать]. Их можно использовать аналогично анимациям 'Idle' [Безделье] и 'Walk' [Ходьба], вот и попробуйте их!

» **Через месяц** Игра *Висельник Чед* почти готова, но страдает немотой – нужен звук.



Hardcore Linux Проверьте себя, участвуя в сложных проектах для продвинутых пользователей.

Compiz: Пробуем

Джонатан Остин продемонстрирует, как получить самые лучшие эффекты из Vista, OS X и более, слегка посидев на общественной ветке проекта Compiz!



Compiz – это клево. Вы, небось, и сами это знаете – уже заставили себя посмотреть ролик с колеблющимися полупрозрачными окнами (просто для самоутверждения) или вращали под музыку куб с рабочими столами (как я?). Однако если вы обошли любовью установку Compiz после того, как Дэвид Ривмэн [David Reveman] из Novell выпустил исходный код в феврале, считайте себя в пролете.

Когда руководство по созданию расширений для этого чудо-OpenGL ускоренного оконного менеджера появилось в сети, на форумах Ubuntu замелькали заплатки, добавляющие новые возможности к Compiz. QuinnStorm, 23-летняя программистка из Питтсбурга, известная своим большим вкладом в исходный код, упорядочила этот хаос, добавив новые возможности, применив заплатки других людей и собрав все в пакет. Ее работа стала общественной веткой Compiz под названием ‘QuinnSource’.

Список возможностей *Compiz-Quinn* слишком длинен, чтобы привести его здесь – но, действуя по инструкциям на последующих страницах, вы получите размытые заголовки окон, собственные оконные рамки и список «Что сделать» поверх вашего куба; раскидаете окна по рабочим столам и отслоите их, чтоб узнать, что под ними; или подключите D-BUS к Compiz – пусть делает всякие трюки за вас; да много чего еще. Но хватит капать слюной: пора за дело.



Наш эксперт

Джонатан Остин уже не один месяц изыскивает время, чтобы пописать код для Compiz. Он автор логотипа Compiz, а также глава команды по разработке темы Blubuntu.

Часть 1 Установим Compiz-Quinn

QuinnStorm сама выпускает пакеты для Ubuntu (работающие на Dapper и Edgy); существуют также RPM-пакеты для SUSE Factory и Fedora, их выпускают Cyberorg и Sams – см. первую часть руководств по каждому дистрибутиву в этом разделе, чтобы узнать подробности.



› **«Отслоим» окно с форумами Compiz и увидим, что под ним скрывается.**

Обычно эти пакеты достаточно свежи, но поклонники истинной свободы делают сборку из исходных текстов. К счастью, это несложно, если вы работаете на Ubuntu (6.06), SUSE или Fedora. Просто следуйте руководству, чтобы учесть все зависимости для вашего дистрибутива, а затем перейдите к разделу Сборка, чтобы собрать *Compiz-Quinn*.

Данные рекомендации предполагают, что у вас уже установлен *Compiz* – если нет, то смотрите списки HOWTO для вашей системы и графической карты: в эту статью все тонкости не уместятся. Поддерживается ли ваша карта или нет, узнайте на http://gentoo-wiki.com/HARDWARE_Video_Card_Support_Under_XGL. Прежде чем что-либо установить, подумайте: большинство описываемых здесь программ пока что в находится разработке и в любой момент может измениться, сломаться или исчезнуть в клубах дыма (увы, я не имею в виду спецэффект при закрытии окна).

Ubuntu

Начнем с того, что отредактируем **sources.list** и включим репозитарии сообщества:

```
sudo gedit /etc/apt/sources.list
```

Теперь добавьте одну из следующих трех строк (это зеркала):

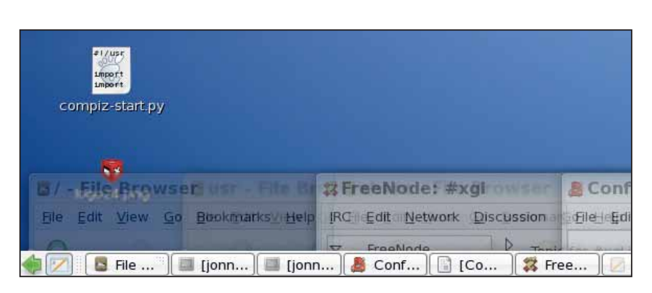
```
deb http://media.blutkind.org/xgl/ dapper main
deb http://ubuntu.compiz.net/ dapper main
deb http://www.beerorkid.com/compiz/ dapper main
```

а затем

Месяц назад Как использовать расширение Greasemonkey для настройки сайтов «на лету».



эффекты



› Подобно Моисею, расширение Show Desktop заставляет море окон расступиться, и вы попадаете на свой рабочий стол.

```
deb http://xgl.Compiz.info/ dapper main
```

Далее, возьмем GPG-ключ QuinnStorm и добавим его в *APT*, чтобы обновить базу пакетов и получить самые свежие пакеты:

```
$ wget http://www.beerorkid.com/Compiz/quinn.key.asc -O - |
  sudo apt-key add -
sudo apt-get update
sudo apt-get install Compiz cgwd Compiz-gnome xserver-xgl
  cgwd-themes
```

Если собирать пакеты вам не надо, переходите к Части 2. А если вы собираете *Compiz*, боюсь, придется установить куда больше пакетов. Возможно, некоторые из них уже у вас есть, если вы собирали программы раньше – я разместил список, достаточный для меня, в файле **Magazine/Compiz/Text/Ubuntu Build** Deps на диске к журналу, и вы можете просто скопировать его и вставить. Список покажет вам, сколько нового кода вы затащите в систему. Не советую за это браться, если вы добиваетесь стабильности и боитесь что-то поломать!

SUSE

На нашем уроке я буду использовать *Smart* в качестве менеджера пакетов SUSE. Если у вас еще не стоит *Smart*, можете найти его на странице http://snipurl.com/vfw2 или просто найти RPM-пакет для вашей архитектуры на диске в каталоге **Magazine/Compiz/Utilities/smart**. Обзаведясь им, можно начать установку программ. Наберите **#smart update** для настройки каналов (вдобавок, сейчас самое время поставить Smart-gui).

Теперь добавьте репозитарии *Compiz-Quinn* в источники Smart:

```
smart channel --add http://software.openSUSE.org/download/
  compiz-quinn/SUSE_Linux_10.1/compiz-quinn.repo
sudo smart update
```

Если вы используете версию 10.2 Beta или Factory, замените **SUSE_LINUX_10.1** на **SUSE_Factory**.

Теперь мы можем установить последние версии всех программ:

```
sudo smart install compiz-quinncvs xgl-cvs cgwd-themes gsetcompiz
```

Эта команда обновит ваши программы до последних версий. На время написания этой статьи новейшие расширения не загружались автоматически, поэтому вам придется загружать их вручную с помощью *Gconf* или использовать команду, чтобы подключить все расширения, которые Quinn добавила в версию Ubuntu – текст находится в **Text/AllPlugins**.

```
gconftool-2 -s /apps/compiz/general/allscreens/options/active
  plugins -t list --list-type=string [gconf,decoration,reflection,blur,
  obbly,fade,minimize,cube,rotate,zoom,scale,move,resize,place,db
  s,switcher,water,trailfocus,state,neg,bs,showdesktop]
```

Однако этот набор может измениться – могут добавиться другие расширения, также необходимые для корректной работы. Кто не хочет собирать версию сам, может спокойно перейти ко 2-й части.

К несчастью, сборка *Compiz-Quinn* для SUSE 10.1 не так проста, как для Ubuntu, потому что для этого нужно скомпилировать свою версию Mesa. *Compiz* требует libGL.so из Mesa версий 6.5 и выше, а SUSE 10.1 все еще поставляется с версией 6.4.1 (с версиями Factory и 10.2 Beta все будет в порядке). Я включу здесь необходимые зависимости для случая, если вы захотите собрать собственную версию Mesa, или для тех, кто будет делать пакеты Mesa 6.5 для 10.1 в ближайшем будущем. Я отметил разделы 'Gnome Development' и 'C/C++ Development' в менеджере пакетов Yast2, и вы их тоже отметьте, как и все прочее из списка **Magazine/Compiz/Text/SUSE Build Deps**. Теперь переходим к разделу Сборка.

Fedora Core

Благодаря Sams из www.illawarra.org/linux у вас уже есть *Compiz*, остается только добавить некоторые новые репозитарии Yum и получить последние версии пакетов. Запустите все последующие команды от лица root (**su -**, а не просто **su**):

```
rpm -ivh http://www.illawarra.org/linux/sam-repo-5-2.fc5.noarch.
  rpm
yum install xorg-x11-server-Xgl sam-switch compiz-quinn cgwd
  cgwd-themes gset-compiz gconf-editor
```

Теперь сбросьте свойства в Gconf, чтобы использовать новые настройки для *Compiz*:

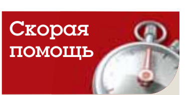
```
export GCONF_CONFIG_SOURCE="xml:merged:/etc/gconf/
  gconf.xml.defaults"
gconftool-2 --makefile-install-rule /opt/Xgl/etc/gconf/schemas/
  compiz.schemas
```

Последнее: запустите от имени суперпользователя **sam-switch**, для настройки новых пакетов. Сохраните все ваши документы перед запуском **sam-switch xgl** (как вам вежливо советует первая команда), X будет остановлен!

Если вы работаете только с пакетами, переходите ко второй части. В противном случае установите зависимости для сборки (в менедже- »



› Trailfocus уменьшает непрозрачность окна, яркость и насыщение, и сразу видны недавно работавшие окна.



Во время написания этой статьи QuinnStorm реорганизовала свои пакеты для Ubuntu на compiz-plugins отдельно от compiz-core, чтобы можно было чаще проводить обновления. Это не влияет на команды в этом HOWTO, но на будущее это полезно знать.



Скорая помощь

С помощью расширения Scale-All, нажав F10, вы увидите все окна со всех рабочих столов.

» ре пакетов я отметил группы Development Libraries, Development Tools, Gnome, X Software Development; сделайте то же самое). Зависимости можно найти на диске в **Magazine/Compiz/Text/Fedora Build Deps**.

Сборка

Если вы хотите собрать самую свежую версию *Compiz-Quinn*, скачайте последнюю версию исходного кода из CVS-репозитория Quinn в подходящий каталог – мы будем использовать **srcdir**. На случай проблем при сборке, мы припасли на диске, в каталоге **Magazine/Compiz/Code**, копию исходных текстов, используемых на данном уроке. Возможно, эта версия уже малость [или даже не малость, – прим. ред.] устарела – ну тогда ждите от CVS исходников, которые согласятся собраться...

Чтобы скачать код из CVS, наберите

```
mkdir srcdir
cd srcdir
cvs -d:pserver:anonymous@metascape.afraid.org:/cvsroot co
   compiz cgwd
```

Когда эта команда выполнится, можно продолжать. На машине, где уже установлен *Compiz*, вы, возможно, захотите использовать **ключ --prefix=<ДРУГОЙ_ПУТЬ>** и **autogen.sh**, чтобы не портить установленную версию. По умолчанию make install помещает все в каталоги **/usr/local/bin** и **/usr/local/lib/Compiz** – это подходит всем наши трем дистрибутивам. Для Fedora и SUSE мы статично компонуем Mesa (v6.5), поэтому добавим в конец **./autogen.sh** строки соответственно для Fedora и SUSE:

```
--with-gl-libs="/lib/libGL.a -pthread -ldrm"
--with-gl-libs="/usr/lib/libGL.a -pthread -ldrm"
```

Вот код:

```
cd compiz
./autogen.sh --enable-libsvg-cairo --disable-kde
make
sudo make install
```

Теперь соберем Cgwd (Custom Generic Window Decorator). На этот раз для Fedora используйте **PKG_CONFIG_PATH=/usr/local/lib/pkgconfig/**, перед запуском **./autogen.sh**.

```
cd ..
cd cgwd
./autogen.sh
make
sudo make install
```

Этот код поместит новый двоичный файл *Compiz* в **/usr/local/bin**, а Cgwd – в **/usr/bin/cgwd**. В Debian и Ubuntu **/usr/bin/Compiz** – не двоичный файл, а скрипт, устанавливающий правильное значение LD_PRELOAD для различных карт. Чтобы скомпилированный вами двоичный файл заработал, применяется следующий трюк: переименуйте **/usr/local/bin/Compiz** в **Compizcvs.real** и поместите его в скрипт, аналогичный **/usr/bin/Compiz**. Полезно назвать этот скрипт, например, '*Compiz*cvs': тогда вам не придется указывать полный путь до двоичного файла каждый раз при вызове обычного *Compiz*.

Часть 2 Быстрые эффекты

Строительные работы завершены – можете использовать старый метод для запуска *Compiz* и инициализации версии QuinnStorm. Заметим, что Gnome-window-decorator был замещен на *Cgwd*, поэтому просто замените одну строчку на другую во всех используемых скриптах запуска, и все будет работать. У вас нет старого метода для запуска *Compiz*? Не беспокойтесь, перейдите к разделу «Иконка для сброса *Compiz*».

Пакеты *Compiz-Quinn* уже готовы показать всю свою красу, но вы, видимо, захотите все переделать по своему вкусу. Чтобы настроить *Compiz*, понадобится использовать один из графических инструментов настройки: лучшим на данный момент является *Gconf*. Почти вся функциональность *Compiz-Quinn* реализована в виде расширений, то есть любой эффект можно убрать, просто удалив его из списка в **/apps/Compiz/general/allscreens/options/active_plugins**. Некоторые расширения необходимо загружать в определенном порядке – например, эффект размытия загружается после декораций. Все это указано для каждого расширения в **/apps/Compiz/plugins/<ваше_расширение>**.

Прежде чем продолжать, имеет смысл исправить наиболее общие проблемы, преграждающие путь к нирване трехмерного рабочего стола: в частности, удалить черную «панель об уведомлении запуска» *Metacity*; это делается с помощью

```
Gconftool-2 --set --type boolean /apps/panel/global/enable_
animations "False"
Gconftool-2 --set --type boolean /desktop/gnome/interface/enable_
animations "False"
```

Если у вас нет *Gconftool-2*, можете сами поменять ключи через *Gconf*.

Также хорошо бы прекратить перезапуск X по Shift+Backspace, с помощью

```
xmodmap -e "keycode 22 = BackSpace BackSpace Terminate_
   Server"
```

Если проблема будет всплывать при перезапуске *Compiz*, просто добавьте эту строку в ваш скрипт запуска *Compiz* (или *Compiz*-start.py).

В-третьих, пусть нижняя часть окна исчезнет с экрана –

```
gconftool-2 --set --type boolean /apps/compiz/plugins/move/
   allscreens/options/constrain_y_bottom "False"
```

Наконец, если вы обнаружили, что полноэкранные приложения (например, видео или игры) прерываются или тормозят, смените **/apps/Compiz/general/screen0/options/unredirect_fullscreen_windows** на *true* – это значит, что во время работы полноэкранных приложений *Compiz* будет отдыхать, а вы получите прирост в скорости.



› **Движки тем Cgwd преподнесут вам необычные и прикольные эффекты, как эта тема от Zootreeves. Художники, помните: с властью приходит и ответственность....**

Эффекты

На этом этапе урока я могу расслабиться и дать вам заняться исследованием. Позабавьтесь – следующий список эффектов покажет вам, какие классные вещи вы можете делать еще до начала настройки:

» Бросайте свои окна

Включите **/apps/Compiz/plugins/wobbly/screen0/options/release_effect** и установите **/apps/Compiz/plugins/wobbly/screen0/options/release_friction на '0.8'**. Теперь, если вы перетаскиваете окно и отпускаете кнопку мыши, продолжая двигать мышь, то можете «перебросить» окно на другой рабочий стол.



› Этот эффект не сложнее смены изображения небосвода в расширении Cube. Картинка крутится вместе с кубом!

» Покажите все окна в переключателе Эффект временно восстанавливает минимизированные окна при использовании переключателя (Alt+Tab), и вы можете бегло осмотреть их. Просто установите **/apps/Compiz/plugins/switcher/screen0/options/temp_unminimize в** *on*. **Или воспользуйтесь /apps/Compiz/plugins/switcher/screen0/options/auto_rotate**, чтобы увидеть все открытые окна (прыгать по ним можно при помощи комбинацией клавиш Alt+Ctrl+Tab).

» Небосвод Добавьте статичное или анимированное фоновое изображение в пустое пространство за кубом, добавив путь к квадратному изображению или изображению размером степени двойки (то есть 1024x1024 или 1024x2048) в **/apps/Compiz/plugins/cube/screen0/options/skydome_image** и установив *skydome* и/или *skydome_animated* в *true*.

» Эластичное изменение размера Установите **/apps/Compiz/plugins/resize/allscreens/options/stretch_texture** в *on*, чтобы *Compiz* увеличивал текстуру при изменении размера окна, когда вы отпустите мышь.

» Вращайте куб при достижении границы экрана мышью Для этого установите **/apps/Compiz/plugins/rotate/allscreens/options/edge_flip_pointer** в *on* и выберите значения по вкусу для **/apps/Compiz/plugins/rotate/allscreens/options/flip_time**.

» Новые способы просмотра окон Движения мыши в различные углы рабочего стола инициируют различные эффекты перемещения и отображения всех ваших окон. По часовой стрелке: показать только окна текущего приложения, показать рабочий стол (спрятать окна) и, наконец, собрать окна со всех рабочих столов. Измените значение **/apps/Compiz/plugins/scale/screen0/options/organic** – увидите более естественную раскладку окон.

» Масштабируйте окна И наконец, я выделяю этот пункт, потому что он находится в странном месте: опция для настройки «масштаба» окон проживает в **/apps/Compiz/plugins/minimize/screen0/options/zoom_created_windows**.

Размытие и темы

Видимо, ваш энтузиазм от этих 30-секундных эффектов иссяк, и я покажу, как применить некоторые новые эффекты к заголовкам окна и прочему.

Прозрачность идеальна для окон дома, но на компьютере она может сделать неотличимыми задний и передний фон на рабочем столе. К счастью, есть несколько расширений размытия. Одно из самых быстроразвивающихся расширений сделал, большей частью, некто Idontknow12. Главная его особенность – опция размытия без использования пиксельных шейдеров (**/apps/Compiz/plugins/blur/allscreens/options/mode**), поэтому даже люди вроде меня с GeForce 4 MX могут получить прикольные эффекты, рассчитанные на более мощное оборудование.

Расширение Idontknow12 может размывать все, что *Compiz* способен сделать прозрачным, установив **/apps/Compiz/plugins/blur/allscreens/options/transparency_blur** в *on*; вместе с Trailfocus это выглядит замечательно. Если хотите, можете поменять настройку расширения, чтобы размывать все подряд, с помощью *Gconf*.

Граница под контролем

Другое крутое расширение от Idontknow12 – Отражение [Reflection]: выберите PNG-изображение, и оно будет растянуто и наложено на заголовок окна (независимо от того, есть ли размытие). Сделать собственную карту размытия в Gimp очень просто: создайте полупрозрачный PNG с дизайном, который вы хотите видеть на границах окна, например, полосы, шаблон или градиент (один из них помещен в каталог **Utilities** на диске). Поместите изображение где угодно, а затем вставьте полный путь до него в **/apps/Compiz/plugins/reflection/allscreens/options/reflection_map**.

Но это еще не все – с помощью *Cgwd* и *GCompizthemer* от QuinnStorm можно назначать темы для заголовков окон. *Cgwd* применяет различные «движки» для эффектов, таких уже есть несколько штук – движок 'pixmap' самый гибкий и мощный и создает поистине вызывающие темы. Благодаря Cornelious, кнопки освещаются при наведении на них курсора, и можно сделать внешний вид максимально приближенным к Vista.

Для создания собственной темы, подберите PNG-файл для каждой кнопки и добавьте его в разрастающийся *GCompizthemer* (самый простой способ научиться – просмотреть темы в **Cgwd-themes**). Можно взять любую их тем, присланных пользователями.

Приоденем окна

Любителей организовывать окна и придавать им большую узнаваемость порадует расширение State: оно позволяет устанавливать и хранить яркость, насыщение, прозрачность и первоначальный порт просмотра любого окна при загрузке. Для каждого свойства у него отдельные ключи, поэтому вы просто вписываете списки строк, разделенных запятыми. Строки выглядят таким образом: <тип_фильтра>:<значение_фильтра>:<значение>. В вашем арсенале – следующие типы фильтров:

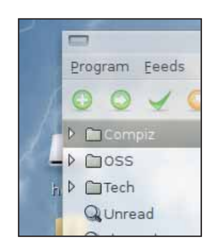
» w Тип окна. Принимает значения Dock, Toolbar, Menu, Utility, Splash, Normal, Dialog, ModalDialog или Unknown. Используйте xprop | grep TYPE и посмотрите последнее слово в строке, чтобы узнать тип любого окна.

» c Класс окна. Обычно совпадает с именем исполняемого файла, только первый символ написан в верхнем регистре. Вы можете узнать класс окна с помощью xprop | grep WM_CLASS.

» t Заголовок окна. Имя окна, отображаемое в его заголовке.

» p Имя исполняемого файла, того же, что и исполняемый файл в **$PATH**.

Для упражнения, добавьте 'c:Gnome-terminal:80' в **/apps/Compiz/plugins/state/screen0/options/opacity**, чтобы у всех ваших окон терминала была прозрачность *80*, а если вы добавите 'w:Menu:95' и 'w:Unknown:95', то ваши меню и всплывающие подсказки еще более похорошеют. »



› Только стеклянные окна должны быть прозрачными. Настоящие компьютеры размывают фон, чтобы зрелище было более спокойным.



› Vista? Нет, спасибо, Билл. Это на карте без поддержки пиксельных шейдеров и с 64 МБ ОЗУ.

Сюрпризы от кодеров Compiz

» Используя Show Desktop, нажмите F7, чтобы убрать с дороги все окна.
От PlayerX

» Используя Blurred Titlebars, можно изменить в Gconf значение размытия и карту отражения.
От Idontknow12

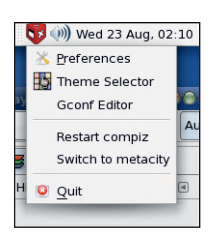
» Рост и сжатие: Окна появляются в месте положения курсора мыши и сжимаются при разрушении окна.
От QuinnStrom

» С помощью Neg, обращайте цвета окон (Super+N) или всего экрана (Super+M).
От Moppsy

» Эффект Water нарисует рябь и дождь на вашем экране (Ctrl+Super, Shift+F8 вызывает дождь).
От Дэвида Ривмэна

Скорая помощь

Если у вашей видеокарты есть пиксельные шейдеры, можете увеличить значение размытия и повысить качество до 'best', для более интенсивного размытия. Включите transparency_blur, а off_onopacity установите в *off*, для полного размытия.



› **Потому что случаются проблемы...**

Часть 3 Внедримся на рабочий стол

Здесь показано несколько трюков и секретов, которые позволят *Compiz* укорениться в вашем рабочем процессе и на рабочем столе (в случае с *Xwinwrap* – буквально). Дочитав этот раздел, вы будете считать *Compiz* не заменой *Metacity* или Kwin, но интегрируемым инструментом, без которого нельзя обойтись.

Xwinwrap был одним из инструментов, которые Дэвид Ривмэн (David Reveman) демонстрировал при выпуске *Compiz*; он позволяет поместить видео или хранитель экрана на рабочий стол как анимированный фон. Если его не нашлось в вашем репозитарии, то скомпилировать его несложно и самому:

```
cd srcdir
cvs -d :pserver:anoncvs@cvs.freedesktop.org:/cvs/xapps co Xwinwrap
cd Xwinwrap
make
sudo cp Xwinwrap /usr/bin/Xwinwrap
```

Официальной документации вроде бы нет, но Moppsy (участник разработки *Compiz*) выловил следующее из исходного кода:

'-g = parse geometry, -ni = no input, -argb = argb visual, -fs = full screen, -s = sticky, -st = skip taskbar, -sp = skip pager, -a = above, -b = below, -nf = no focus -o OPACITY'

Набрав **Xwinwrap**, вы получите справку по использованию: *Xwinwrap* вам все про себя расскажет, если его запустить без аргументов командной строки. Вот для начала несколько примеров:

- Чтобы запустить хранитель экрана, наберите строку

```
Xwinwrap -ni -argb -fs -s -st -sp -a -nf -- /usr/lib/xscreensaver/
  glmatrix -root -window-id WID
```

- А чтобы запустить видео, наберите

```
Xwinwrap -ni -o 0.6 -fs -s -st -sp -b -nf -- MPlayer -wid WID -quiet
  movie.mpg
```

- Чтобы зациклить короткое видео, скажем, зеленые луга, вы можете использовать -loop <число> (0 – бесконечный цикл) как аргумент для *MPlayer*. Если у вас достаточно новая версия *Rxvt* или *Urxvt*, можно даже установить его на рабочем столе:

```
Xwinwrap -ni -argb -fs -s -st -sp -b -nf -- urxvt -depth 32 -bg
  rgba:0000/0000/0000/cccc -fg white
```

Иконка для сброса Compiz

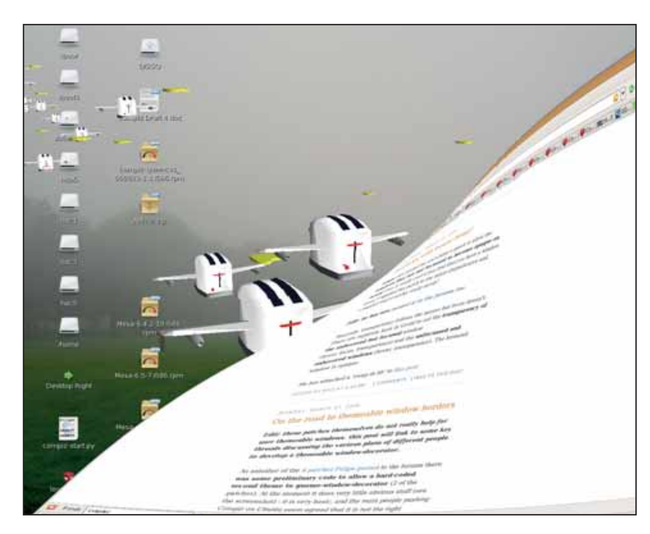
Compiz не всегда работает так гладко, как вам хочется – поэтому, когда все работает как у Боба из Microsoft и вы остаетесь без оконного менеджера, полезно иметь иконку для перезапуска *Compiz* или замены его на *Metacity*. Скрипт на Python от Gandalfm, *Compiz-start.py*, выполняет эту работу, а также предоставляет доступ к функциям *Compiz* и инструментам настройки.

Чтобы попользоваться волшебством от Gandalfm, возьмите код из каталога **Magazine/Compiz/Utilities/Compiz-start0.3** (можете пропустить эту часть, если используете пакеты SUSE: они поставляются со своей собственной версией). В этом каталоге сделайте следующее:

```
chmod 755 Compiz-start.py
sudo mv Compiz-start.py /usr/bin/
sudo mv logo24.png /usr/share/Compiz
```

и вы готовы к запуску (то есть к выполнению *Compiz-start.py*). Вы можете добавить скрипт в автозагрузку сессии, используя каталог автозапуска KDE или закладку Startup в диалоге Session configuration в Gnome. Чтобы он работал, не забудьте удалить все предыдущие методы автозапуска. Заметим, что для работы всех опций у вас должны быть установлены *gCompizthemer* и *gset-Compiz* – удалить их вызовы вы всегда сможете, отредактировав скрипт. Если вы не используете *Cgwd*, измените все вызовы 'cgwd' в скрипте на 'gnome-window-decorator'.

Я добавил в код несколько закомментированных «шаблонов» для новых пунктов меню. Не забудьте добавить определение команды и пункт меню, с помощью которого ее можно вызвать.



› **Это что, тостеры? С Xwinwrap любой рабочий стол движется – попробуйте видео, хранители экрана или даже терминал.**

Расширение с D-BUS

Наш следующий подвиг – поделиться *Compiz* с другими скриптами и программами при помощи D-BUS, системы, позволяющей приложениям общаться друг с другом. Интеграция D-BUS с *Compiz* доставит множество хранителей экрана, программ запуска и приложений, использующих функции *Compiz*: например, вращение куба, переключение окон и установка прозрачности. Такая интеграция позволит разработчикам проделывать то же, что и с Aplle Quartz Extreme – расширение Miniwin служит хорошим примером.

Например, я удалил переключатель рабочих столов со своей панели, обнаружив, что он плохо работает с моими 8 рабочими столами (можете выбрать любое число до 32 в **/apps/Compiz/general/screen0/options/size**). Вместо этого я использую две иконки по краям моей панели для вращения рабочих столов в любом направлении. Если вам тоже этого хочется, сначала создайте новую кнопку запуска на панели, назовите ее Следующий Стол или как вам угодно и назначьте иконку (эта кнопка будет вращать куб вправо, чтоб вы знали, какую выбрать стрелку). Теперь во вкладке Выполнить наберите код:

```
dbus-send --type=method_call --dest=org.freedesktop.Compiz
  org/freedesktop/Compiz/rotate/allscreens/rotate_right org
  freedesktop.Compiz.activate string:'root' int32:0x52 string:'x'
  int32:[X_BUTTON_POS] string:'y' int32:[Y_BUTTON_POS]
```

заменив **X\Y_BUTTON_POS** на координаты кнопки. (Вы можете найти команду в **Text\DBUS Commands** в пакетах поддержки и две программы запуска в **Utilities** – просто не забудьте изменить координаты экрана.)

Повторите то же самое для другой стороны панели, изменив rotate_right на rotate_left и задав координаты для кнопки. Все!

Фактически модуль DBUS позволяет вам вызвать все, что определено в коде как 'action' [действие], то есть (для начинающих) выполнить почти все, на что можно назначить кнопку. Официальная документация по этому поводу ограничивается комментарием в **dbus.c**, поэтому заставить что-то работать как надо не всегда просто.

Для начала поглядим на **srcdir/Compiz/plugins/dbus.c** и **srcdir/Compiz/plugins/rotate.c**. Как вы увидите из второго блока комментариев в верху **dbus.c**, Дэвид Ривмэн подтверждает, что можно вызвать любое действие. Заголовки у действий следующие:

```
static Bool
rotateLeft (CompDisplay *d,
CompAction *action,
CompActionState state,
CompOption *option,
int nOption)
```



» «Док» и первая версия Miniwin. Новая версия будет использовать D-BUS, и приложения типа kiba-dock смогут создавать «мини-окна».

Чтобы выполнить эти действия, дадим *Compiz* весточку через *dbus-send*:

```
dbus-send --type=method_call --dest=org.freedesktop.Compiz
    org/freedesktop/Compiz/rotate/allscreens/rotate_left org
    freedesktop.Compiz.activate string:'root' 0x52 string:'x' int32:1280
    string:'y' int32:1024
```

Разберем сообщение: **--type** определяет тип сообщения; **--dest**, понятное дело, говорит dbus-send, куда посылать сообщение.

Далее мы передаем «путь объекта назначения» (то есть «путь» до действия). Действия вызываются по именам, указанным в коде – первые несколько строк объявления действия для «повернуть влево» выглядят так:

```
o = &rd->opt[ROTATE_DISPLAY_OPTION_LEFT];
o->name = "rotate_left";
o->shortDesc = N_("Rotate Left");
o->longDesc = N_("Rotate left");
o->type =
CompOptionTypeAction;
o->value.action.initiate = rotateLeft;
```

Итак, имя мы получаем из o->name, а полный «путь» будет выглядеть как **/org/freedesktop/Compiz/<имя_расширения>/allscreens/<имя_действия>**.

Следующая строка из команды *dbus-send,* описывающая вращение куба, это тип сообщения (который всегда org.freedesktop.Compiz.activate), за которым следуют пары ключ\значение, описанные Дэвидом в комментарии. При повороте куба налево мы используем три из этих пар – первое дает нам идентификатор окна 'root', который всегда равен 82 (или 0x52 в шестнадцатиричной системе) при использовании Xgl; другие два – значения x и y курсора мыши после поворота. Я установил их равными значениям позиции кнопок, чтобы мышь появилась в том месте, где вы ее оставили. Более точно, мы посылаем строку 'x', чтобы *Compiz* понял, что это имя переменной и целое значение 1270, в которое необходимо установить 'x'.

Чтобы узнать, какие посылать переменные, посмотрите код; просмотрите функцию **getIntOptionNamed**, используемую для вызова желаемого действия (это значение **o->value.action.initiate** в объявлении). В **rotateLeft** внутри **rotate.c** мы видим

```
o[0].value.i = getIntOptionNamed (option, nOption, "x", pointerX);
```

что можно классифицировать как установку переменной в значение x положения курсора. Вы увидите похожие вызовы для 'root' и 'y'.

Не просто развлечение

Теперь вы можете вращать куб и получить больше рабочих столов, но, возможно, пустое пространство на верхней грани куба не дает вам покоя – «Должен же я его использовать...» Что ж, можете поместить там SVG-изображение или их список, чтоб устроить слайд-шоу. Mattdev121 употребил *Gconf* и поместил туда список дел.

Для этого последнего трюка не нужен никакой новый код *Compiz*, просто хорошая идея и скрипт *Bash*. В Ubuntu вам, правда, понадобится еще *librsvg2-bin*. Скопируйте папку **cubetodo** из **Magazine/Compiz/Utilities** в ваш домашний каталог (если хотите поместить ее куда-нибудь в другое место, то придется редактировать скрипт). Откройте терминал и наберите

```
cd ~/cubetodo
```



» Забейте на список дел и наслаждайтесь вращением кубика...

```
sed -e 's/width=\"1280\"/width=\"YOUR_X_RES\"/g' -e 's
    height=\"1024\"/height=\"YOUR_Y_RES\"/g' template_base.svg >
    template.svg
```

заменив **YOUR_X\Y_RES** на соответствующие значения.

Запустите скрипт, набрав **./CubeToDo.sh**. Появится *Nano*, куда можно вписать ваш список неотложных дел. Для его запоминания нажмите Ctrl+O (+Enter), а затем Ctrl+X для выхода – полюбуетесь в терминале на надпись 'working', и после короткой паузы изображение на кубе изменится. В каталоге **cubetodo** находится программа запуска, которую можно добавить на панель или в меню *Compiz-start.py*.

Что дальше?

Думаете, ваш рабочий стол уже битком набит? Но разработка *Compiz* продолжается. Появляются интересные расширения, обсуждаемые на форумах, например, Copacity, который меняет определенный цвет в окне на прозрачный, позволяя «устаревшим» приложениям отлично сочетаться с другими. Также ведутся разговоры о проекте Agora Project для создания среды использования 3D-ускорения на рабочем столе. Есть надежда создать новые эффекты и поднять уровень функциональности до Aplle Core Video/Image/Animation в Linux.

Пока OS X и Vista кричат о 3D, *Compiz* атакует рабочие столы Linux: реорганизует, изменяет, растягивает, перерисовывает и вращает их. В этой революции Linux находится на верном пути. LXF

Наше спасибо...

Разработчикам: QuinnStorm, iXce, playerX, moppsy, throughnothing, Idontknow12, cyberorg, gandalfn, MacSlow, Michael Callahan, reggaemanu, zootreeves, cornelius, sparX, Athropos, David Reveman и приятным людям из Novell, Ddave, PsyberOne, g14, tommie-lie, Melvil, Jesper, сообществу compiz.net. Этот список неисчерпаем!

Снимки экранов: Who, PlayerX, profoX, Capso.

Ресурсы

Настройте свою карту в любимом дистрибутиве
» http://en.opensuse.org/Using_Xgl_on_SUSE_Linux
» http://fedoraxgl.tuxfamily.org/index.php?title=Installation_en
» http://ubuntuforums.org/showthread.php?t=148351
» http://gentoo-wiki.com/HOWTO_XGL
» http://forum.mandrivaclub.com/viewtopic.php?t=47071&highlight=xgl

Другие системы: http://wiki.compiz.net

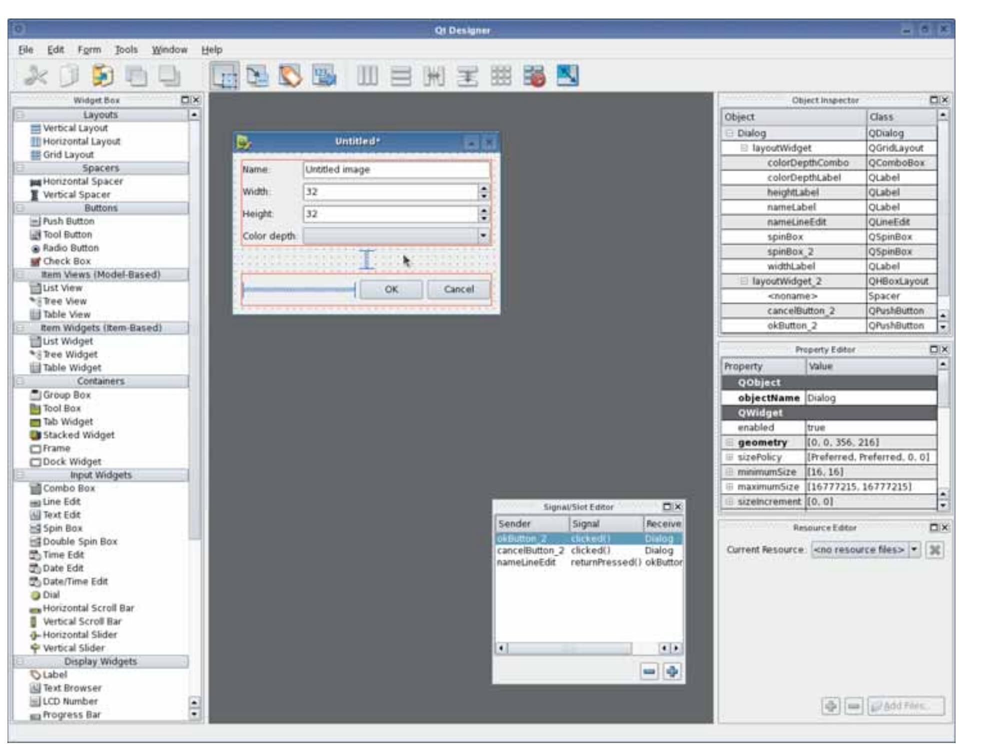
Сообщество Compiz и информация:
» http://compiz.blogspot.com
» http://compiz.net
» http://wiki.compiz.net
» #compiz-dev на irc.freenode.net

Будущие разработки:
» www.compiz.net/topic-2197-looking-forminiwin-testers
» www.compiz.net/topic-1644-copacityplugin
» http://agora.underdev.org/index.php

» **Через месяц** Построим систему вещания с архитектурой «клиент-сервер» с Python и Kamaelia.

Компоненты

ЧАСТЬ 8: В заключение этой длинной серии **Андрей Боровский** даст вам несколько советов по написанию компонентов KParts.



Целое всегда больше суммы своих частей
Аристотель

Серия статей, посвященных программированию Qt и KDE, подходит к своему финалу, и вполне логично, что в заключительной статье мы рассмотрим написание компонентов *KParts*. Это можно считать хорошим финалом нашей серии не только потому, что программирование *KParts* – одна из самых сложных тем, но еще и потому, что сами компоненты *KParts* представляют собой венец развития многих технологий KDE. Например, система *XMLGUI*, которая изначально задумывалась только как средство настройки интерфейса пользователя, нашла неожиданно удачное применение в рамках *KParts*. В детской истории, ни названия, ни автора которой я, к сожалению, не помню, главный герой плавал по морям и все время носил подтяжки. Подтяжки эти пригодились ему только один раз, когда он встретился... с кашалотом, и в этот момент они спасли ему жизнь (увы, я не помню, что именно отважный мореход сделал с кашалотом). Такие элементы KDE, как *XMLGUI* и *KAction* напоминают подтяжки из рассказа. Они кажутся ненужными до тех пор, пока программист не встретится с кашалотом, то есть с компонентами и модулями. Материал этой статьи может служить отправной точкой для изучения программирования некоторых специализированных типов компонентов, таких, например, как компоненты *KOffice Parts*. Если вы задумаете программировать компоненты *KOffice*, вам придется набраться терпения, потому что эти компоненты реализовать сложнее, чем рассмотренные в этой статье обычные компоненты *KParts*.

Все пользователи KDE сталкивались с компонентами *KParts*, хотя не все, возможно, знают об этом. Каждый раз, когда вы щелкаете мышью по значку графического файла (или файла другого поддерживаемого типа) в окне *Konqueror*, файл-менеджер переходит в режим просмотра. При этом *Konqueror* загружает компонент *KParts*, способный отобразить содержимое файла данного типа. В режиме просмотра меняется и интерфейс браузера – в строке меню и на панели быстрого доступа появляются команды компонента-просмотрщика. Откуда берутся все эти компоненты *KParts*? Хотя компонент может существовать и сам по себе, как правило, он является частью какого-либо самостоятельного приложения. Например, для просмотра растровой графики *Konqueror* использует компонент GVImage_part, который не является частью какого либо приложения (по крайней мере, в моей системе SUSE 10.1 нет приложения gvimage), тогда как для просмотра документов в формате PDF используется компонент kpdf_part, который является основой самостоятельного приложения *KPDF*. Следует подчеркнуть, что приложение, основанное на компоненте KParts, обычно взаимодействует с ядром этого компонента напрямую, минуя оболочку *KParts*. *KParts* – стандартизированный интерфейс, предназначенный для взаимодействия между частями разных приложений, которые не учитывают внутренних особенностей друг друга. Так, например, приложение KPDF не использует функциональность *KParts* для взаимодействия с компонентом kpdf_part, хотя такое взаимодействие, в принципе, возможно. Интерфейс *KParts*, предоставляемый компонентом kpdf_part, используют другие приложения, например, *Konqueror*. На практике все это означает, что если у нас есть какое-либо приложение KDE, мы можем превратить его в источник компонента *KParts*, ничего не меняя в его основе. Само превращение самостоятельной программы в компонент *KParts* выполняется очень просто, что мы сейчас и увидим.

Почему компоненты *KParts* создаются на основе самостоятельных приложений? Во-первых, таким образом расширяется сфера их применения. Зачастую более функционально иметь и приложение, и соответствующий ему компонент. Пользователь может работать с приложением, поставляющим *KParts*, как с самостоятельным инструментом или как с компонентом, загружаемым другой программой. Во-вторых, отлаживать самостоятельные приложения проще, нежели отлаживать компоненты *KParts*. Обычно сначала пишут и отлаживают программу, а затем на ее основе создают компонент, добавляя интерфейс *KParts* к уже проверенному в работе ядру приложения. Мы тоже

Месяц назад Мы изучали модули KDE.





пойдем этим путем и создадим компонент *KParts* на основе программы *images*, которую мы рассматривали в позапрошлой статье. Если у вас нет позапрошлого номера журнала, – не беда, исходные тексты программы *images* (не модифицированные) вы найдете на диске журнала в фале **images.tar.gz**. На основе программы *images* мы создадим компонент images_part, который сможет открывать графические файлы для просмотра и выполнять над ними простые преобразования (исходные тексты приложения с компонентом вы найдете в файле **images_part.tar.gz**).

Как и рассмотренные в предыдущей статье модули KDE, компоненты KParts экспортируются из разделяемых библиотек, поэтому разработку нашего компонента мы начнем с того, что преобразуем основную часть программы *images* в разделяемую библиотеку. Как вы помните (а если не помните, перечитайте пятую часть этой серии), главным визуальным элементом программы *images* является класс imagesView. Этот класс не только отображает содержимое графического файла, но и содержит методы, выполняющие преобразование изображения. Наша задача заключается в том, чтобы вынести класс imagesView в разделяемую библиотеку. Сделать это очень просто, нам даже не придется менять ни одной строчки исходных текстов *images*. Откройте в текстовом редакторе файл **Makefile.am** (тот, который лежит в **images/src** в старом варианте и в **images-part/src** – в новом). Файл **Makefile.am** используется инструментом *autoconf* для генерации Make-файла и содержит описание основных параметров целевого приложения. Ниже приводится фрагмент этого файла, который нас сейчас интересует (для сокращения листинга комментарии удалены):

```
bin_PROGRAMS = images images_client
INCLUDES        = $(all_includes)
images_LDFLAGS = $(KDE_RPATH) $(all_libraries)
images_LDADD   = $(LIB_KFILE) $(LIB_KDEPRINT)
images_SOURCES = main.cpp images.cpp imagesview.cpp pref.cpp
```

Заменим этот фрагмент следующим:

```
lib_LTLIBRARIES = libimages1.la
bin_PROGRAMS = images
libimages1_la_SOURCES = images.cpp imagesview.cpp pref.cpp \
imagesiface.skel   SettingsDialog.ui settingsdialogimpl.cpp images_part.cpp
libimages1_la_LDFLAGS = $(all_libraries) -version-info 1:0:0 -module
libimages1_la_LIBADD = $(LIB_KFILE) $(LIBVM) -lKParts
INCLUDES        = $(all_includes)
images_LDFLAGS = $(KDE_RPATH) $(all_libraries)
images_LDADD   = $(LIB_KDECORE) libimages1.la
images_SOURCES = main.cpp
```

Строка

```
lib_LTLIBRARIES = libimages1.la
```

указывает, что мы добавляем в проект *images* новую библиотеку. В моей системе уже есть библиотека *libimages*, так что я добавил к имени цифру 1. Далее мы сообщаем, что теперь большая часть исходных текстов исполняемого файла **images** становится частью библиотеки *libimages1* (имена файлов исходных текстов библиотеки содержит переменная libimages1_la_SOURCES). Сравните значение, присваиваемое libimages1_la_SOURCES в новом варианте файла и значение, присваиваемое images_SOURCES в старом варианте. Переменная libimages1_la_LIBADD содержит список библиотек, от которых зависит библиотека *libimages1*. Сюда «перешли» все библиотеки, от которых зависела программа *images*, плюс библиотека *libKParts*, которая понадобится нашему компоненту *KParts*. Сама программа *images* ссылается теперь только на библиотеку *libimages1* (переменная **images_LDADD**), а ее исходные тексты состоят из одного файла **main.cpp**. Теперь мы можем заново сгенерировать Make-файл для нашего приложения и пересобрать его. Новый вариант программы *images* работает так же, как и прежний, но теперь программа устроена по-другому. Основная ее часть теперь содержится в библиотеке *libimage1,* а исполняемый файл **images** играет роль оболочки для этой библиотеки.

Следующим шагом нужно добавить в библиотеку *libimages1* оснастку, превращающую класс imagesView в компонент KParts images_part. В библиотеку *libimages1* мы добавим реализацию трех классов, обеспечивающих работу компонента. Если эти классы покажутся вам очень сложными, не отчаивайтесь – их реализация почти не зависит от специфики компонента, для которого они предназначены. Создавая новый компонент, вы можете просто перенести в него код этих классов, после чего нужно будет изменить в нем лишь несколько строк. Добавим в проект библиотеки два новых файла – **images_part.h** и **images_part.cpp** (в среде *KDevelop* это можно сделать с помощью Automake Manager). В файле **images_part.h** объявим три класса:

```
class ImagesFactory : public KLibFactory
{
  Q_OBJECT
public:
  ImagesFactory();
  virtual ~ImagesFactory();
  static KInstance *instance();
  static KAboutData *aboutData();
protected:
  QObject* createObject( QObject* parent = 0, const char* name = 0,
                         const char* className = "QObject",
                         const QStringList &args = QStringList() );
private:
  static KInstance *s_instance;
};
class ImagesPart: public KParts::ReadOnlyPart
{
  Q_OBJECT
public:
  ImagesPart(QWidget *parent, const char *name);
  virtual ~ImagesPart();
protected:
  virtual bool openFile();
  virtual bool closeURL();
protected slots:
   void proctoBlacknWhite();
private:
  imagesView *widget;
  ImagesBrowserExtension *m_extension;
  KAction * actiontoBlacknWhite;
};
class ImagesBrowserExtension : public KParts::BrowserExtension
{
```

»

```
»  Q_OBJECT
   friend class ImagesPart;
public:
   ImagesBrowserExtension(ImagesPart *parent);
   virtual ~ImagesBrowserExtension();
};
```

Каждый компонент *KParts* должен создать два класса – класс-фабрику объектов *KParts* и класс, инкапсулирующий функциональность *KParts*. Класс, реализующий фабрику объектов, должен происходить от класса KLibFactory, а класс KParts – либо от класса KParts::ReadOnlyPart, либо от класса KParts::ReadWritePart. Классы, происходящие от KParts::ReadOnlyPart соответствуют компонентам, предназначенным для просмотра (но не модификации файлов). Классы-потомки KParts::ReadWritePart соответствуют компонентам, реализующим функции редактирования. Главным классом нашего компонента является класс ImagesPart, происходящий от класса KParts::ReadOnlyPart.

Рассмотрим подробнее класс-фабрику ImagesFactory. В файле **images_part.cpp** должна быть определена функция init_libimages1(), которая создает объект класса-фабрики объектов:

```
extern "C"
{
   void *init_libimages1()
   {
      return new ImagesFactory;
   }
};
```

Имя этой функции состоит из префикса init_ и имени библиотеки, экспортирующей компонент (приложение, загружающее нашу библиотеку для того, чтобы получить компонент images_part, ожидает найти в ней функцию с таким именем). Когда приложению понадобится создать компонент images_part, оно вызовет эту функцию и получит указатель на объект ImagesFactory, который затем и создаст главный объект компонента. Основная задача объекта класса-фабрики заключается в том, чтобы не допустить появления двух объектов класса ImagesPart. Поскольку приложение ничего не знает о классе ImagesFactory, оно будет работать с соответствующим объектом как с объектом базового класса KLibFactory. Соответственно, мы должны перекрыть в классе ImagesFactory виртуальные методы, унаследованные им от классов-предков. Метод ImagesFactory::createObject(), объявленный как чистый виртуальный, создает объект класса ImagesPart. В предыдущих версиях KDE эту функцию выполнял метод create(). Метод instance() либо создает объект класса KInstance, либо возвращает ссылку на уже созданный объект этого класса. Объект класса KInstance позволит нашему компоненту получить доступ к тем глобальным объектам KDE, которые обычно доступны самостоятельным приложениям. Как уже отмечалось, вы можете просто скопировать весь код класса-фабрики в проект своего компонента. Для того, чтобы этот код работал в вашем проекте, нужно изменить только имя главного класса компонента, который создает метод createObject().

Перейдем теперь к конструктору класса ImagesPart:

```
ImagesPart::ImagesPart(QWidget *parent, const char *name)
  : KParts::ReadOnlyPart(parent, name)
{
   setInstance(ImagesFactory::instance());
   QWidget *canvas = new QWidget(parent);
   canvas->setFocusPolicy(QWidget::ClickFocus);
   setWidget(canvas);
   m_extension = new ImagesBrowserExtension(this);
   actiontoBlacknWhite = new KAction(i18n("To &Black && White"),
                  QIconSet(BarIcon("blacknwhite")), 0, this,
                  SLOT(proctoBlacknWhite()),
                  actionCollection(), "bw_command");
   actiontoBlacknWhite->setToolTip(i18n("Transforms image palette to
grayscale"));
   setXMLFile("images_part.rc");

   widget = new imagesView(canvas);
   widget->show();
}
```

В конструкторе мы инициализируем поля этого класса. Обратите внимание на объект canvas класса QWidget. С помощью метода setWidget() мы указываем приложению, загружающему компонент, что объект canvas является главным визуальным элементом компонента. На самом деле объект canvas невидим, но он будет родителем нашего «настоящего» главного визуального элемента в иерархии графических элементов. Пустой объект класса QWidget нужен для того, чтобы «настоящий» визуальный элемент компонента мог сам управлять своим размером. Дело в том, что когда *Konqueror* (или другое приложение) создаст экземпляр нашего компонента, его главный визуальный элемент (тот, который зарегистрирован в приложении-хозяине с помощью метода setWidget()) автоматически будет «растянут» на все главное окно приложения. Чтобы избежать нежелательных эффектов, мы передаем приложению не «настоящий» графический объект, а «пустой» объект canvas, и получаем возможность самостоятельно контролировать размеры «настоящего» визуального элемента. Указатель на «настоящий» главный визуальный элемент, – объект класса imagesView, хранится в поле widget.

Класс ImagesBrowserExtension – потомок класса KParts::BrowserExtension, должен способствовать более тесной интеграции компонента, основанного на ReadOnlyPart и приложения-браузера. Браузеры KDE обладают способностью работать с сетевыми ресурсами как с локальными файлами, и класс KParts::BrowserExtension помогает основному классу компонента обрабатывать URL. Наш компонент способен загружать только локальные файлы, так что мы не заботимся о поддержке сетевых URL.

Пропустим пока остальные элементы конструктора и рассмотрим метод openFile():

```
bool ImagesPart::openFile()
{
   QPixmap pm(m_file);
   widget->setPixmap(pm);
   widget->setAutoResize(true);
   return !pm.isNull();
}
```

Этот метод перекрывает виртуальный метод класса-предка. Приложение-хозяин вызывает метод openFile() для того, чтобы компонент мог открыть соответствующий файл. Имя файла (точнее, его URL) хранится в этот момент в поле m_file. Как и в самом приложении *images*, мы загружаем изображение с помощью объекта класса QPixmap. Метод openFile() должен возвращать значение *true*, если ему удалось открыть URL, и *false* в противном случае (мы проверяем результат операции с помощью метода isNull()). Метод closeURL(), который также объявлен в базовом классе, нужен для тех компонентов, которые поддерживают работу с сетевыми ресурсами. Этот метод вызывается приложением, если нужно прервать текущий процесс загрузки сетевого ресурса (это может понадобиться, например, в том случае, если пользователь отменил загрузку ресурса или начал загружать другой ресурс, не дожидаясь завершения загрузки предыдущего).

Добавим кнопочки

В принципе, у нашего компонента теперь есть все, что нужно для того, чтобы отображать содержимое файла, однако большинство компонентов, даже тех, что предназначены только для просмотра, этим не ограничивается. Как правило, у компонентов есть визуальные элементы управления, которые позволяют контролировать процесс просмотра. Например, стандартный компонент просмотра изображений предоставляет пользователю команды увеличения, уменьшения и поворота, а также команды перехода к следующему и предыдущему файлам. Мы тоже добавим один элемент управления в компонент images_part. Как вы помните, приложение *images* может модифицировать открытое изображение, в частности, конвертировать пол-

ноцветную палитру в палитру оттенков серого. Эту команду мы и добавим в наш компонент. Вернемся к конструктору ImagesPart. В этом конструкторе мы создаем объект-действие actiontoBlacknWhite. Мы уже создавали объекты-действия в седьмой и пятой частях этой серии, так что останавливаться подробно на создании объекта actiontoBlacknWhite мы не будем. Наш объект-действие связан со слотом proctoBlacknWhite(), который выполняет команду преобразования палитры:

```
void ImagesPart::proctoBlacknWhite()
{
  widget->toBlacknWhite();
}
```

Вы уже знаете, что для того, чтобы добавить новый элемент интерфейса, недостаточно создать объект-действие. Нужно еще и указать приложению, где и как отображать соответствующие элементы меню и панели быстрого доступа. В этом нам, как всегда, поможет технология *XMLGUI*. Обратите внимание на вызов метода setXMLFile() в конструкторе ImagesPart. Этот метод указывает имя файла, из которого компонент должен взять описание интерфейса. Добавим в проект сам файл **images_part.rc**:

```
<!DOCTYPE kpartgui>
<kpartgui name="images">
<MenuBar>
    <Menu name="Processing"><text>&amp;Proc    essing</text>
      <Action name="bw_command"/>
    </Menu>
  </MenuBar>
  <ToolBar name="Images-ToolBar">
    <Action name="bw_command"/>
  </ToolBar>
</kpartgui>
```

Мы указываем, что команда преобразования изображения должна располагаться в подменю Processing главного меню (такого подменю в *Konqueror* нет, значит, оно будет создано автоматически). В этом файле нет ничего, что было бы нам не знакомо, за исключением, пожалуй, тэга kpartgui, который указывает, что мы имеем дело с компонентом *images*.

Мы почти закончили программирование нашего компонента. Однако, если мы установим его сейчас, то не увидим никаких изменений. Дело в том, что среда KDE по-прежнему ничего не знает о нем. Мы должны еще отредактировать файл **images.kdelnk**, который описывает наше приложение *images*. В этот файл следует добавить три строчки:

```
MimeType=image/jpeg;image/gif;image/bmp;image/png
ServiceTypes=Browser/View
X-KDE-Library=libimagess1
```

Первая строчка указывает типы MIME, с которыми может работать наше приложение *images* и соответствующий ему компонент. Мы могли не указывать эти данные, когда программа *images* была простым самостоятельным приложением, однако их необходимо указывать для компонента, иначе браузер *Konqueror* просто не будет знать, когда следует отображать соответствующие компоненту элементы управления. Между прочим, теперь *Konqueror* будет знать, что не только компонент images_part, но и приложение *images* способно работать с графическими файлами и добавит в свои меню соответствующие команды. Вторая строчка указывает, какие сервисы предоставляет наш компонент. В данном случае это сервисы просмотра содержимого файлов указанных типов. Третья строчка указывает имя библиотеки компонента. Теперь, наконец, мы можем установить компонент images_part с помощью *make install*. Запустите экземпляр файл-менеджера *Konqueror* после установки компонента. В окне браузера щелкните правой кнопкой мыши значок какого-нибудь файла одного из поддерживаемых типов. В открывшемся контекстном меню выберите команду «Просмотреть в *images*». В результате выбранный файл должен быть открыт для просмотра с помощью нашего компонента (рис. 1).

Как видите, *Konqueror* добавил новое подменю в главное меню и кнопку на панель быстрого доступа.

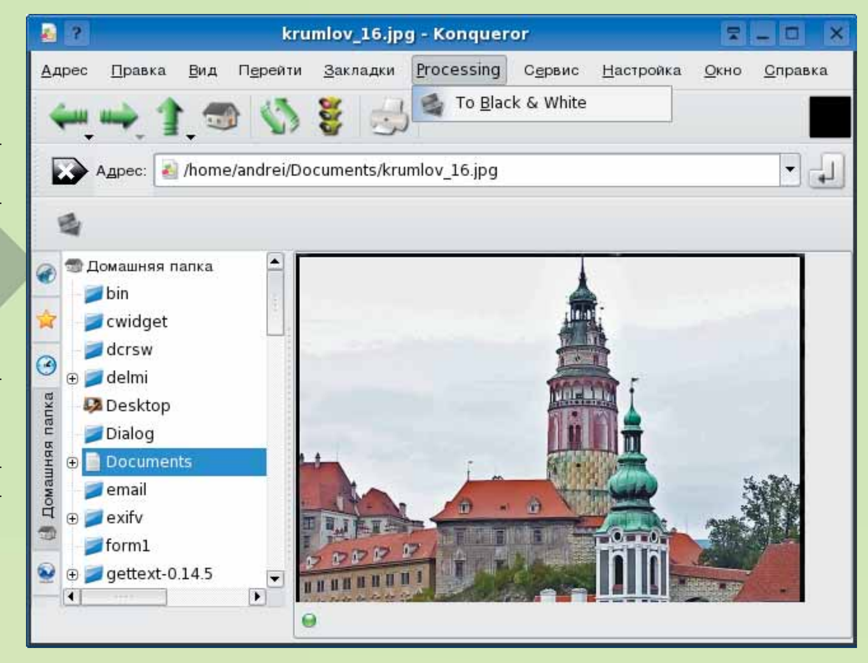


Рисунок 1. Компонент images_part в работе *KParts*.

Финальный аккорд

Компонент images_part предназначен только для просмотра изображений. Скажем несколько слов о компонентах, допускающих редактирование. Как уже упоминалось, главный класс компонента-редактора должен быть потомком класса KParts::ReadWritePart. Помимо уже известных нам методов openFile() и closeURL(), в этом классе объявлен ряд новых виртуальных методов, предназначенных для перекрытия в классах-потомках. Самым важным из этих методов является метод saveFile(), который должен выполнять сохранение файла, открытого в компоненте. Этот метод работает так же, как и openFile(), то есть приложение-хозяин вызывает его, когда пользователь дает соответствующую команду. Вы также, возможно, захотите перекрыть метод setReadWrite(), который переключает компонент между состояниями «только для чтения» и «чтение и запись».

***

На этом мы, как я уже обещал, заканчиваем серию статей по программированию Qt и KDE. Надеюсь, что теперь вы сможете оценить возможности этой технологии и, главное, дать квалифицированный ответ на вопрос, может ли Qt/KDE служить основой для вашей собственной разработки. LXF



Процессы

ЧАСТЬ 6: В процессе написания этой статьи **Андрей Боровский** чуть было не породил армию злобных зомби. Оказывается, для того чтобы стать дипломированным некромантом, не обязательно семь лет учиться в Хогвартсе...

Все процессы должны быть у нас под контролем

М.С. Горбачев





Понятие процесса играет ключевую роль в современных ОС. Существует много определений процессов; я воспользуюсь весьма простым, данным в [1]: процесс – это выполняющийся экземпляр программы. Хотя это определение, в общем, «работает», оно не совсем подходит к многопоточным программам Linux, о которых мы поговорим в следующей статье. Важный аспект процессов с точки зрения ОС – неделимость процесса относительно назначения ресурсов. Система выделяет ресурсы процессу в целом, а не его частям. Внутри процесса задача управления ресурсами ложится на программиста. Для каждого процесса ядро системы поддерживает специальную структуру данных task_struct, в которой хранятся важнейшие параметры процесса (процессы уровня пользователя не имеют доступа к этой структуре). В структуре task_struct есть специальное поле, в котором хранится численный идентификатор процесса (Process Identifier, PID). Именно этот идентификатор используется для обозначения процессов на уровне прикладного API Linux.

Под управлением процессами мы будем понимать, прежде всего, создание новых процессов и контроль за их выполнением. Обычно новые экземпляры программы создаются с помощью вызова функции fork(2), с которой мы уже многократно встречались в статьях этой серии. Помимо функции fork() Linux предоставляет нам еще один, весьма тонкий инструмент – функцию clone(2). Эта функция позволяет настраивать разделение ресурсов между процессами, но не умеет раздваивать процесс в точке вызова, как это делает fork(). Специально для того, чтобы вас запутать, существует еще и системный вызов sys_clone(), который умеет раздваивать процессы в точке вызова. Вы, конечно, знаете, что, будучи вызвана один раз, функция fork() возвращает управления дважды, двум копиям исходного процесса (родительской и дочерней). Поскольку вы знаете также, что эти две копии процесса похожи как две капли воды, будет полезнее перечислить ниже не сходства этих процессов, а их различия.

- Родительскому процессу функция fork() возвращает PID дочернего процесса, а дочернему процессу – значение *0*. Именно по возвращенному fork() значению процесс может узнать, дочерний он или родительский. В иерархии процессов родительский процесс оказывается родителем (странно, не правда ли) дочернего процесса. Дочерний процесс может получить PID родительского процесса с помощью вызова getppid(2), в то время как родительский процесс может узнать PID своего дочернего процесса только из результата fork(). Именно поэтому fork() может позволить себе не возвращать значение PID дочернему процессу, но обязана возвращать значение PID дочернего процесса родительскому.
- Значения tms_utime, tms_stime, tms_cutime, и tms_cstime в дочернем процессе обнуляются.
- Блокировки на уровне файлов (file locks), установленные в родительском процессе, в дочернем процессе снимаются (иначе сразу два процесса оказались бы владельцами блокировок).
- Сигналы, обработка которых была отложена в родительском процессе в момент вызова fork(), в дочернем процессе сбрасываются.

Если вы хотите запустить из вашей программы экземпляр другой программы, вы должны пожертвовать для этого уже существующим процессом. Если вы хотите, чтобы новая программа работала одновременно со старой, нужно сначала раздвоить процесс с помощью fork(), а затем заместить образ программы в одной из копий образом новой программы. Для замены образа одной программы образом другой применяются функции семейства exec*. Функций этого семейства

Месяц назад Мы изучали сигналы Unix.



и потоки

всего шесть: execl(), execlp(), execle(), execv(), execvp() и execv(). Если вы присмотритесь к названиям функций, то увидите, что первые три имеют имена вида execl*, а вторые три – имена вида execv*. Эти функции отличаются списком параметров, а также тем, как обрабатывается имя файла.

Рассмотрим, например, функции execve(2) и execvp(2). Прототип функции execve(2) имеет вид:

```
int execve(const char *pathname, char *const argv[], char *const envp[]);
```

Первый параметр – это имя исполняемого файла запускаемой программы (им может быть двоичный файл или сценарий оболочки, в котором в первой строке указан интерпретатор при помощи конструкции #!). Имя исполняемого файла для этой функции должно включать полный путь, (т.е. путь, начинающийся с корневого слэша, точки или тильды). Второй параметр функции представляет собой список аргументов командной строки, которые должны быть переданы запускаемой программе. Формат этого списка должен быть таким же, как у списка argv[], который получает функция main(), то есть первым элементом должно быть имя запущенной программы, а последним – нулевой указатель на строку, то есть

```
(char *) NULL
```

В последнем параметре функции execve() передается список переменных среды окружения в формате

```
ИМЯ=ЗНАЧЕНИЕ
```

Массив переменных должен заканчиваться нулевым указателем на строку, как и argv. Копию набора переменных окружения, полученного вашей программой от оболочки, можно получить из внешней переменной environ, которую вы должны объявить в файле вашей программы:

```
extern char ** environ;
```

Передача списка переменных окружения явным образом позволяет вам, в случае необходимости, модифицировать список. В параметре argv, как и в параметре envp, можно передавать значения *NULL*, если вы уверены, что вызываемой программе не нужны переменные окружения или командная строка. Заголовок функции execvp() выглядит проще:

```
int execvp(const char *file, char *const argv[]);
```

Первый параметр – это имя запускаемого файла. Функция execvp() выполняет поиск имен с учетом значения переменной окружения PATH, так что для программы, которая расположена в одной из директорий, перечисленных в PATH, достаточно указывать одно лишь имя исполняемого файла. Второй параметр функции соответствует второму параметру execve(). У функции execvp() нет параметра для передачи набора переменных окружения, но это не значит, что запущенная программа не получит эти переменные. Программе будет передана копия набора переменных окружения родительского процесса.

Вы можете изменить переменные среды окружения не только путем модификации списка environ, но и с помощью набора специальных функций. Для установки новой переменной окружения используется putenv(3). У функции putenv() один параметр типа char*. В этом параметре функции передается пара ИМЯ=ЗНАЧЕНИЕ для установки новой переменной. Функция возвращает *0* в случае успеха и *-1* в случае ошибки. Программа может прочитать значение переменной окружения с помощью вызова getenv(3). У этой функции так же один параметр типа «строка ASCIIZ». В этом параметре передается имя переменной, значение которой мы хотим прочитать. В случае успеха функция возвращает строку, содержащую значение переменной, а в случае неудачи (например, если запрошенной переменной не существует) – *NULL*.

Переменные среды окружения играют важную роль в работе процессов, поскольку многие системные функции используют их для получения различных параметров, необходимых для нормальной работы программ; однако управление программами с помощью переменных окружения считается морально устаревшим методом. Если ваша программа должна получать какие-то данные извне, воздержитесь от создания новых переменных. Современные программы чаще полагаются на файлы конфигурации, средства IPC и прикладные интерфейсы. Старайтесь также не использовать переменные окружения для получения тех данных, которые вы можете получить с помощью специальных функций C: например, используйте getcwd(3) вместо обращения к переменной PWD. Полный список установленных переменных окружения и их значений вы можете получить с помощью команды env, а некоторые наиболее важные переменные приведены в таблице:

Переменная	Описание
DISPLAY	Имя дисплея X Window
HOME	Полный путь к домашней директории пользователя-владельца процесса
HOST	Имя локального узла
LANG	Текущая локаль
LOGNAME	Имя пользователя-владельца процесса
PATH	Список директорий для поиска имен файлов
PWD	Полный путь к текущей рабочей директории
SHELL	Имя оболочки, заданной для пользователя-владельца процесса по умолчанию
TERM	Терминал пользователя-владельца процесса по умолчанию
TMPDIR	Полный путь к директории для хранения временных файлов
TZ	Текущая временная зона

Помимо функций getenv() и putenv(), есть еще несколько функций для работы с переменными среды окружения. Функция setenv() позволяет установить значение переменной. В отличие от функции putenv(), эта функция позволяет передать флаг, благодаря которому значение уже существующей переменной не будет изменено. С помощью функции unsetenv() вы можете удалить переменную окружения. Наконец, если переменные окружения вам надоели, вы можете воспользоваться функцией clearenv() для уничтожения их всех скопом. Под удалением здесь подразумевается удаление переменных из среды окружения





текущего процесса и его потомков. Это никак не повлияет на переменные окружения других процессов.

Рассмотрим использование функции execvp() на примере программы, запускающей другую программу, имя которой передается ей в качестве аргумента командной строки. Назовем нашу программу *exec* (ее исходные тексты вы найдете на прилагаемом диске в файле **exec.c**). Если вы скомпилируете файл **exec.c** и скомандуете, например

```
./exec ls -al
```

программа выполнит команду ls -al и возвратит сведения о том, как был завершен соответствующий процесс. Ниже приводится полный исходный текст *exec*.




```
#include <sys/types.h>
#include <sys/wait.h>
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>

int main(int argc, char * argv[])
{
  int pid, status;
  if (argc < 2) {
    printf("Usage: %s command, [arg1 [arg2]...]\n", argv[0]);
    return EXIT_FAILURE;
  }
  printf("Starting %s...\n", argv[1]);
  pid = fork();
  if (pid == 0) {
    execvp(argv[1], &argv[1]);
    perror("execvp");
    return EXIT_FAILURE; // Never get there normally
  }
  else {
    if (wait(&status) == -1) {
      perror("wait");
      return EXIT_FAILURE;
    }
    if (WIFEXITED(status))
      printf("Child terminated normally with exit code %i\n",
WEXITSTATUS(status));
    if (WIFSIGNALED(status))
      printf("Child was terminated by a signal #%i\n",
WTERMSIG(status));
    if (WCOREDUMP(status))
      printf("Child dumped core\n");
    if (WIFSTOPPED(status))
      printf("Child was stopped by a signal #%i\n", WSTOPSIG(status));
  }
  return EXIT_SUCCESS;
}
```

Поскольку мы хотим, чтобы новая программа не заменяла старую, а выполнялась одновременно с ней, мы раздваиваем процесс с помощью fork(). В дочернем процессе, которому fork() возвращает 0, мы выполняем вызов execvp(). Первый параметр execvp() – значение argv[1], в котором должно быть передано имя запускаемой программы. В качестве второго параметра функции передается массив аргументов командной строки, полученных программой *exec*, начиная со второго элемента (элемент argv[1]). Например, если список аргументов программы *exec* имел вид **exec**, **ls**, **-al**, то первым параметром функции execvp() будет строка ls, а вторым параметром – массив из двух строк **ls** и **-al** (не забывайте, что первым элементом массива аргументов командной строки должно быть имя самой программы, по которому она была вызвана). Таким образом вы можете, например, давать команды

```
./exec ls -al
./exec ./exec ls -al
./exec ./exec ./exec ls -al
```

и так далее. Чего вы не можете, однако, сделать, так это выполнить с помощью нашей программы команду типа

```
./exec "ls > log.txt"
```

Перенаправление ввода-вывода выполняется командной оболочкой. Для того, чтобы выполнять такие команды, необходимо сначала запустить оболочку, например:

```
exec sh -c "ls -al > log.txt"
```

Сразу после вызова функции execvp() в нашей программе следует вывод сообщения об ошибке. Вам может показаться странным, что мы не проверяем значение, возвращенное execlp(), но на самом деле этого и не требуется. Если вызов функции успешен, текущая программа будет заменена другой, и, стало быть, при нормальном завершении exec() следующая за ней инструкция никогда не будет выполнена. Если программа перешла к инструкции после вызова execvp(), значит, заменить образ программы не удалось, и нам остается только вывести сообщение об ошибке.

Обратимся теперь к родительскому процессу. Мы хотим, чтобы родительский процесс дождался завершения дочернего процесса и вывел сообщение о том, как был завершен дочерний процесс. Первую часть этой задачи мы могли бы решить с помощью уже известного нам механизма сигналов. Каждый раз, когда дочерний процесс приостанавливается или завершается, родительский процесс получает сигнал SIGCHLD. Мы могли бы приостановить работу программы до получения этого сигнала, но таким образом мы бы знали только то, что один из дочерних процессов программы завершился, но не знали бы, ни как он завершился, ни какой именно это был процесс. Причина этого, конечно, в том, что сигналы сами по себе не несут никакой дополнительной информации. Однако в нашем распоряжении есть функция wait(2), которая приостанавливает процесс до тех пор, пока один из его дочерних процессов не будет остановлен или не завершится, после чего возвращает информацию о том, какой процесс завершился и что стало причиной его завершения. Значение, возвращаемое функцией wait() – это PID завершившегося процесса, а аргументом функции должен быть указатель на переменную status типа int. В этой переменной функция вернет дополнительные сведения о том, как завершился процесс. Вы могли подумать, что после того как мы создали два процесса с помощью fork(), не так уж важно, запускаем ли мы новую программу в дочернем или в родительском процессе, ведь разница между ними невелика. Теперь вы знаете как минимум одну причину придерживаться строгих правил – ведь родительский процесс может следить за дочерним, в то время как обратное невозможно.

Значение переменной status, в которой функция wait() передает дополнительные данные о завершившемся процессе, представляет собой маску из нескольких разных параметров. В файле **<sys/wait.h>** определены макросы, упрощающие «расшифровку» этой маски. Макрос WIFEXITED возвращает значение *1*, если процесс завершился «добровольно», то есть в результате вызова exit() или _exit(). В этом случае с помощью макроса WEXITSTATUS можно узнать код завершения, возвращенный процессом. Макрос WIFSIGNALED возвращает *1*, если выполнение процесса было завершено сигналом. Номер этого сигнала можно узнать с помощью макроса WTERMSIG. Макрос WIFSTOPPED возвращает значение *1*, если выполнение процесса было приостановлено сигналом, номер которого возвращает макрос WSTOPSIG.

Помимо функции wait(), в нашем распоряжении есть еще две функции, позволяющие приостановить процесс в ожидании изменения состояния дочерней программы. Это функции waitpid(2) и waitid(2). Функция waitpid() может приостановить процесс в ожидании изменения состояния определенного дочернего процесса, заданного значением PID. У функции waitpid() три параметра. Первый параметр – значение PID процесса, завершения которого ждет функция. Если передать в этом параметре значение *-1*, функция будет ждать изменения состояния любого процесса, аналогично wait(). Если первый параметр равен нулю, waitpid() ждет завершения любого процесса из той же группы, что и текущий. Второй параметр waitpid() аналогичен параметру wait(). Третий параметр позволяет указать дополнительные флаги функции.

Например, если установить флаг NOHANG, функция вернет управление немедленно, даже если ни один дочерний процесс не завершился (подробнее об этом будет сказано ниже). Результатом функции waitpid(), так же как и в случае wait(), является идентификатор завершившегося процесса. Функция waitid(), появившаяся в Linux начиная с версии ядра 2.6.9, позволяет установить более тонкий контроль над параметрами ожидаемого события и получить более подробную информацию о сигнале, вызвавшем изменение состояния дочернего процесса. Для большинства задач возможности этой функции явно избыточны.

Исследуя работу программы *exec*, мы еще не коснулись одного важного момента, о котором часто забывают при отладке программ, создающих несколько процессов. Неявно мы все время предполагали, что вызов wait() в родительском процессе произойдет до завершения дочернего процесса. Чаще всего так и будет. Но что произойдет, если дочерний процесс завершится раньше вызова wait() в родительском процессе, например, вследствие ошибки при вызове execvp()? Программируя несколько потоков или процессов, мы всегда должны учитывать подобные варианты развития событий. В нашем конкретном случае беспокоиться не о чем. Если дочерний процесс завершится до завершения родительского процесса, система сохранит его «след», именуемый «зомби». Зомби будет существовать до тех пор, пока его родительский процесс не вызовет wait() (waitpid(), waitid()), или пока родительский процесс не завершится. Система сохраняет процессы-зомби специально для того, чтобы родительский процесс мог исследовать причины завершения дочернего процесса; однако это не всегда удобно. Рассмотрим, например, процесс init, который является «папой» всех пользовательских процессов сеанса Linux. Поскольку init существует на протяжении всего сеанса, любой из его дочерних процессов по завершении оставался бы в виде зомби до тех пор, пока init не вызвал бы одну из функций wait*. Еще один процесс, который может породить множество зомби, это демон... Иногда я должен напоминать себе, что пишу статью по программированию, а не сценарий фильма ужасов. Для того, чтобы процесс не оставлял зомби, можно вызывать функцию waitpid() периодически (с флагом NOHANG, чтобы она не блокировала вызвавший процесс, если в системе нет зомби). Можно поступить и иначе, назначив сигналу SIGCHLD обработчик, в котором вызывалась бы функция wait(). Именно так поступает программа *nozombies*, которую вы также найдете на диске.

```
int main(int argc, char * argv[])
{
  int i, pid;
  struct sigaction sa;
  sa.sa_handler = child_handler;
  sigaction(SIGCHLD, &sa, 0);
  for (i = 0; i < 10; i++) {
    pid = fork();
      if (pid == 0) {
        printf("I will leave no zombie\n");
        exit(0);
      }
      else
        printf("Created a process with the PID %i\n", pid);
    }
  while (1)
    sleep(1);
  return EXIT_SUCCESS;
}
```

Программа nozombies устанавливает обработчик сигнала SIGCHLD (функция child_handler()):

```
void child_handler(int i)
{
  int status;
  wait(&status);
}
```

Далее программа создает несколько дочерних процессов, каждый из которых выводит диагностическое сообщение и завершает свою работу. Затем программа *nozombies* переходит в бесконечный цикл (так что завершать ее придется с помощью Ctrl-C). Пока программа находится в бесконечном цикле, вы можете проверить, оставила ли она процессы-зомби. Для этого откройте другой терминал и скомандуйте

```
ps -al
```

Для сравнения, закоментируйте строку

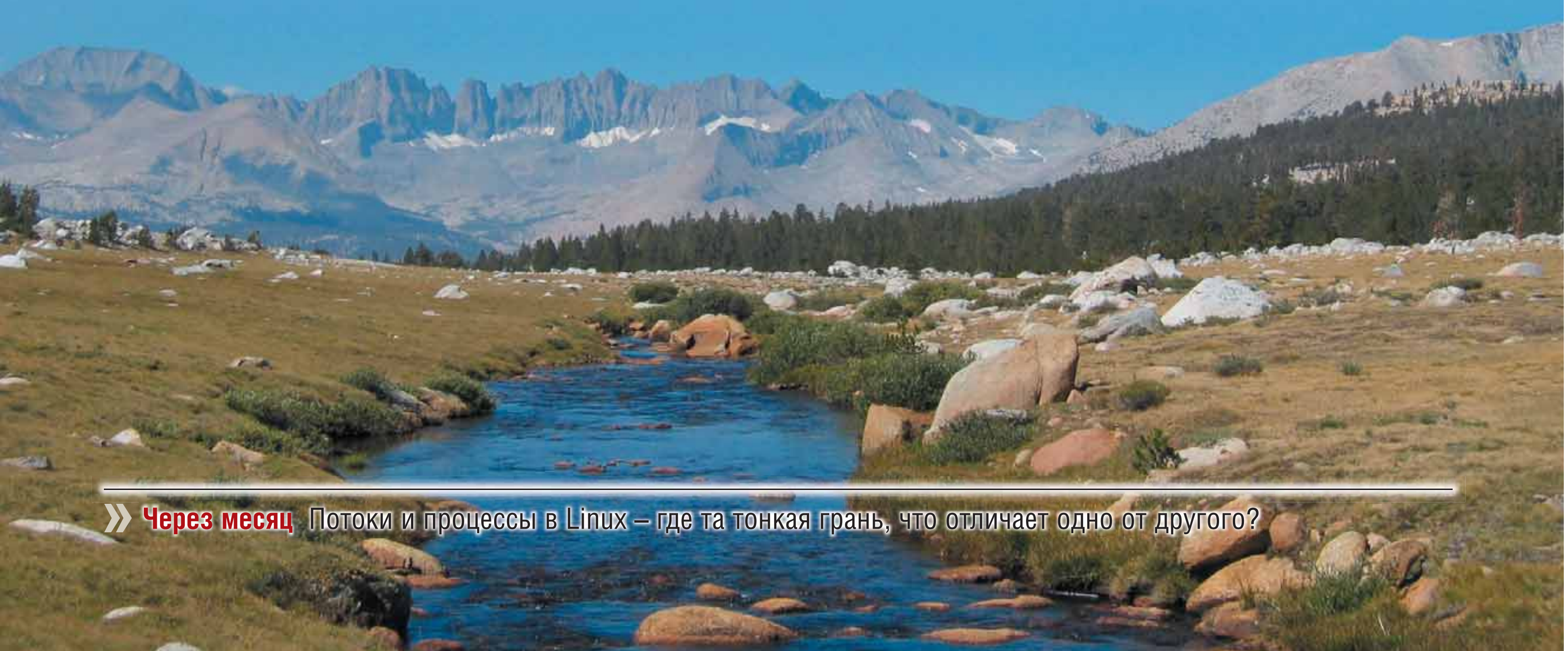
```
wait(&status);
```

в теле функции child_handler(), перекомпилируйте и снова запустите программу. Теперь команда ps –al покажет наличие десяти зомби-процессов (они помечены символом Z).

Интерфейс функций exec* может показаться слишком сложным и, для многих задач, избыточным. Стандарт языка C включает описание функции system(), предназначенной для запуска внешних программ. Единственный аргумент этой функции – строка запуска программы. Поскольку для выполнения программы функция system() запускает копию оболочки (той, которая в вашей системе вызывается командой sh(1)), эта функция выполнит любую команду, которую вы могли бы ввести в командной строке оболочки. Функция system() (пример ее использования вы найдете в файле **system.c**) приостанавливает выполнение вызвавшей ее программы до тех пор, пока дочерний процесс не завершит работу и возвращает код завершения процесса.

Программируя управление процессами, мы часто идем по лезвию бритвы. Нет, дело тут вовсе не в ужасных зомби, а в том, что трудно заставить работать слаженно два независимых процесса. Труднее этого может быть только программирование потоков, которым мы и займемся в следующий раз. LXF





Через месяц Потоки и процессы в Linux – где та тонкая грань, что отличает одно от другого?



ЧАСТЬ 2: Мало сыграть свадьбу – надо еще научиться вести семейный бюджет, выполнять одну и ту же работу каждый день и принимать судьбоносные решения. **Антон Черноусов** продолжает учить монархическую чету премудростям Java.



В предыдущей статье были рассмотрены основы объектно-ориентированного языка программирования Java: инкапсуляция, наследование, полиморфизм. Мы создали простое приложение, в котором женили короля на царевне Несмеяне.

Сегодня мы поговорим о простых типах (числах), о ветвлениях, циклах, а также об исключениях.

Простые числа и их братья

Итак, мы оставили нашу царевну Несмеяну в семейном гнезде, где она в скором времени добралась до казны государства. Ей пришлось освоить нехитрую математику – отнимать и делить: ведь казна требует строго учета, и никакая программа не обходится без вычислений. Без них невозможно создать даже самую простую систему учета и контроля, а мадам необходимо наладить нетривиальный учет бюджетных средств.

Для реализации арифметических операций в Java существует семь арифметических операторов, которые работают с любыми числовыми типами: сложение +, вычитание –, умножение *, деление /, остаток %, унарный минус -val и унарный плюс +val, а также шесть примитивных типов переменных для выполнения арифметических операций:

Тип переменной	Разрядность	Описание
Byte	8	Целое со знаком
Short	16	Целое со знаком
Int	32	Целое со знаком
Long	64	Целое со знаком
Float	32	Число с плавающей точкой
Double	64	Число с плавающей точкой

Значение типов byte и short при выполнении вычислений переопределяется int, поэтому использование этих типов целесообразно для хранения небольших значений для уменьшения объема используемой оперативной памяти.

```
int iA = 10;
byte bB = 12;
int iC = iA + bB;
System.out.println("Сумма A и B = " + iC);
```

Выше приведен пример выполнения арифметической операции с использованием переменных различных типов. Результат – целое число типа int. Для преобразования типа переменной, например, в тип с более низкой разрядностью, необходимо воспользоваться специальными языковыми конструкциями:

```
double dD = 10.2;
double dE = 12.23;
float fF = (float) (dD + dE);
System.out.println(fF);
```

Логика и выбор

Чтобы провести ревизию налоговой службы правильно, Несмеяне пришлось разобраться с налоговым кодексом, а там без поллитра и знаний логических операций не обойтись.

Для представления логических значений в Java присутствует простой тип boolean, который принимает значения *true* или *false*. Для проведения различных логических операций можно использовать ниже перечисленные операторы:

Месяц назад Мы начали изучение Java со старой доброй сказки.

Считалочки

Тип операторов	Операторы				
Операторы отношений	<	>	>=	<=	instanceof
Операторы равенства	==	!=			
поразрядное И	&				
поразрядное исключающее ИЛИ	^				
поразрядное включающее ИЛИ	|				
логическое И	&&				
логическое ИЛИ	||				

Пока Несмеяна разбирается в дебрях налогового кодекса, мы для иллюстрации работы с логическими выражениями обратимся к методу primeNumberCheckUp(int numberForCheck), который выясняет, является ли переданное ему число простым.

```
private boolean primeNumberCheckUp(int numberForCheck) {
    boolean result = true;
    if (numberForCheck != 1 || numberForCheck != 2 || numberForCheck !=
3) {
        for (int currentNumber = 2; currentNumber
< numberForCheck; currentNumber++) {
            if (numberForCheck % currentNumber == 0) {
                result = false;
            }
        }
    }
    return result;
}
```

В primeNumberCheckUp использованы две замечательные конструкции: оператор выбора по условию или условный оператор, и цикл. Условный оператор if имеет следующий синтаксис:

```
if (логическое выражение) {
        операторы группы 1;
}
else {
        операторы группы 2;
}
```

В вышеописанном методе использована упрощенная конструкция (без else), а для вычисления логического выражения использованы операторы: != (не равно) и || (логическое «или»). В полной конструкции в случае истинности логического выражения выполняются операторы группы 1, иначе – группы 2.

Вместо if для управления выполнением кода может быть использована специальная конструкция switch. В рамках этой конструкции вычисляется целочисленное выражение, и в соответствии с полученным результатом ведется поиск соответствующего значения блока case.

```
int flag = 6;
switch (flag) {
    case 1: {
        System.out.println("Царевна в спальне  №1");
        break;
    }
    case 5: {
        System.out.println("Царевна в спальне  №5");
        break;
    }
    default: {
        System.out.println("Кто тут знает, где царевна?");
        break;
    }
}
```

Обратите внимание на оператор break, который обеспечивает немедленный выход из любого блока, а в примере выше прекращает дальнейшую обработку операторов конструкции switch, а также на блок операторов default – этот блок необязательный, он выполняется в случае, если выражение, удовлетворяющие условию, не было найдено среди case.

Циклы

Разобравшись, как все устроено в королевстве, Несмеяна поняла, что все подчиняется принципу «Украл, выпил, в тюрьму» – точнее, во Дворец до следующего месяца. По кругу.

Для реализации «кругового» процесса в Java существует три вида циклов: for, while и do-while. С циклом for мы уже вы встречались в методе primeNumberCheckUp, продемонстрированном ранее, синтаксис этой конструкции следующий:

```
for (инициализация; логическое выражение; приращение) {
        операторы;
}
```

В рамках инициализации производится объявление и установка значений временных переменных. Цикл выполняется до тех пор, пока логическое выражение истинно. На каждой итерации цикла производится приращение временных переменных.

Две другие конструкции циклов можно назвать однояйцевыми близнецами, если бы не маленькое отличие в их использовании. При использования цикла while проверка условия осуществляется до выполнения блока операторов, а в конструкции do-while – после него, то есть действия внутри цикла отрабатывают хотя бы раз. Например, процесс одиночной попойки короля от счастья, что он женат, на языке Java можно описать так: »

```
int i = 100;
while (i > 0) {
    System.out.println(i + " бутылок пива стояло на столе, одну уговорил");
    i--;
}
System.out.println("Все выпил, да!");
int tankard = 1;
int allTankards = 10;
do {
    System.out.println(tankard + " стакан помыт");
    tankard++;
}
while (tankard <= allTankards);
System.out.println("Ну вот и все стаканы помыл");
```

Массивы

Для организации хранения однотипных данных в Java можно использовать массивы. Массивом называется структура данных, которая позволяет хранить несколько значений в одной переменной (определение расплывчатое, но суть передает верно). Объявить массив можно несколькими способами, но я предпочитаю следующий как наиболее логичный:

```
int[] arrayName = new int[length]
```

При этом int – это тип переменных массива, а length – его размерность (число элементов). После того, как массив создан, значения всех переменных равны *null*.

Многомерный массив – это обыкновенный массив, элементами которого являются массивы, причем совсем не требуется, чтобы они имели одинаковую размерность, например:

```
int[][] arrayName = new int[2][];
arrayName[0] = new int[5];
arrayName[1] = new int[2];
```

В некоторых задачах требуется создать массив, состоящий из произвольных чисел. Генерация произвольных чисел возможна благодаря классу Random, который входит в пакет java.util. К слову, пакеты – это иерархически именованные контейнеры, применяемые для изолирования имен классов в некотором пространстве имен, то есть пакеты применяются во избежание конфликтов с именами классов.

Подключение класса Random к программе происходит при использовании зарезервированного слова import и команды import java.util.Random; традиционно объявляемой в начале файла, содержащего разрабатываемый класс.

```
import java.util.Random;
public class RandomGenerator {
  private static Random random = new Random();
  private int base;
  public RandomGenerator(int base) {
    this.base = base;
  };
  public RandomGenerator() {
      this.base = Math.abs(random.nextInt());
  };
  public int getInt(){
    return Math.abs(random.nextInt(this.base));
  };
}
```

Далее, приведенный код позволяет сгенерировать массив псевдослучайных чисел с использованием класса RandomGenerator, код которого приведен выше:

```
int[] newArray = new int[100];
RandomGenerator currentRG = new RandomGenerator(500);
for (int currentNumber = 0; currentNumber < 100; currentNumber++) {
    newArray[currentNumber] = currentRG.getInt();
    System.out.println("newArray["+currentNumber+"] = "+newArray[currentNumber]);
}
```

Иногда невозможно установить точное количество элементов в массиве, в таком случае на помощь приходят динамические массивы ArrayList. ArrayList – это класс, который входит в Collection API. Объекты, созданные на основе ArrayList, более ресурсоемки. Давайте посмотрим на работу динамических массивов на примере задачи по поиску всех простых чисел в диапазоне от 1 до N:

```
private Integer[] getAllPrimeNumbers (int numberForCheck){
  ArrayList primeNumbers = new ArrayList();
  for (int currentNumber = 1; currentNumber <= numberForCheck; currentNumber++) {
    if (this.primeNumberCheckUp(currentNumber)) {
      primeNumbers.add(currentNumber);
    }
  }
  Integer[] result  = (Integer[]) primeNumbers.toArray(new Integer[0]);
  return result;
}

public int[] getIntArray(){
Integer[] arrayOfPrimeNumbers = (Integer[]) GetAllPrimeNumder(this.number);
  int[] result = new int[arrayOfPrimeNumbers.length];
  for (int currentNumber = 0; currentNumber<resault.length; currentNumber++){
    result[currentNumber] = arrayOfPrimeNumbers[currentNumber].intValue();
  }
  return resault;
};

public Integer[] getIntegerArray(){
  return this.GetAllPrimeNumder(this.number);
};
```

В getAllPrimeNumder формируется динамический массив primeNumbers – объект класса ArrayList (для его использования необходимо подключить пакет java.util.ArrayList). Добавление нового объекта в динамический массив производится методом add(Object). После того, как динамический массив сформирован, из него извлекается простой массив методом toArray(Object[]). Обратите внимание, что из ArrayList извлекается массив объектов типа Integer, несмотря на то, что в него помещаются переменные типа int. Для получения массива состоящего из переменных типа int используется метод getIntArray().

Динамические массивы, как правило, используют лишь до тех пор, пока не установлено точное количество элементов, т.к. скорость работы с простым массивом выше и обработка простого массива менее трудоемка.

Для логического завершения работы с простыми числами включим ранее описанные методы (primeNumberCheckUp, getAllPrimeNumder, getIntArray(), getIntegerArray()) в представленный далее класс ArrayOfPrimeNumbers.

```
import java.util.ArrayList;
public class ArrayOfPrimeNumbers {
  private int number;

 public ArrayOfPrimeNumber(int number) {
   this.number = number;
 }
 public int getNumber() {
   return number;
 }
 public void setNumber(int number) {
```

```
        this.number = number;
    }
}
```

Класс, в зависимости от вызываемого метода, возвращает массив, состоящий из элементов int или Integer.

Куча хлама

Вернемся к нашей Несмеяне. Так как королевство ей досталось, прямо скажем, маленькое, пришлось ей еще и за порядком следить, что значит знать, где и что. Здесь ей помог HashMap – такая интересная структура для хранения данных, класс который входит в **java.util**. Экземпляры класса HashMap позволяют хранить различные именованные объекты, которые хранятся в виде пар «ключ – значение». Снаружи объекта типа HashMap нельзя узнать (как в хорошем бардаке), что же там лежит, однако, зная ключ, можно получить то, что ищешь, ну а если ключ неизвестен, то необходимо начать разбирать бардак до тех пор, пока не будет найдено то, что требуется. HashMap часто используется для хранения параметров системы. Как царевна разбирала бардак мужа с помощью HashMap, можно посмотреть ниже.

```
        HashMap sameHeap = new HashMap();
        sameHeap.put("Key1", "Левый носок");
        sameHeap.put("Key2", "Грязная футболка");
        sameHeap.put("Key3", "Правый носок");
        String sameObj = (String) sameHeap.get("Key1");
        System.out.println("Key1 = " + sameObj);
        Map.Entry entry = null;
        Iterator it = sameHeap.entrySet().iterator();
        while (it.hasNext()) {
            entry = (Map.Entry) it.next();
            System.out.println("For key = " + entry.getKey() +
                " value = " + entry.getValue());
        }
```

Для компиляции понадобится подключить java.util.HashMap, java.util.Iterator, java.util.Map – это можно сделать с помощью всего одной строкой вместо трех – **import java.util.*;**.

Как вы можете убедиться, для помещения объекта в HashMap необходимо использовать метод put, где первым параметром будет ключ, а вторым – ассоциируемый с ним объект. Извлечение объекта возможно с помощью метода get при передаче ключа или с помощью последовательного перебора записей в HashMap.

После выполнения кода вы можете увидеть, что элементы в куче лежат не в том порядке, в котором вы их туда кинули (как в жизни, и не только короля). Порядок элементов в коллекции также может меняться во времени. HashMap обеспечивает постоянное время доступа для операций put и get. Как для ключей, так и для элементов допускаются значения типа *null*.

Пятый угол

В хорошо написанной программе, как и в сказочном королевстве, все должно идти своим чередом, но непредвиденные ситуации все равно бывают, так как программа соприкасается с внешним миром (другими программами, файлами, пользователями). Для решения непредвиденных ситуаций в Java существуют исключения и специальные операторы позволяющие отловить и обработать их.

```
try{
    // небезопасный код
} catch(Exception e) {
    // обработка исключительной ситуации
} finally {
    // гарантированно выполняющийся блок
}
```

Выше представлена структура, с помощью которой производится обработка исключительных ситуаций: в блоке try производятся потенциально опасные операции, а в блоке catch производится обработка возникшего исключения (блоков catch может быть несколько). Под блоком подразумевается последовательность операторов, объявлений локальных классов или локальных переменных, заключенных в фигурные скобки. Область видимости локальных переменных и классов ограничена блоком, в котором они определены.

Иногда дополнительно, вне зависимости от того, как будет выполнен блок try, необходимо, чтобы выполнились какие-либо действия (например, освобождение ресурсов), тогда используют оператор finally, который обеспечивает гарантированное выполнение указанного блока операторов.

Исключительная ситуация может быть сгенерирована виртуальной машиной Java, а также программистом. Исключение «выбрасывается» с помощью оператора throw.

```
public float calculate(int sameA, int sameB) throws Exception {
        float result;
        if (sameB != 0) {
            result = sameA / sameB;
        } else
            throw new Exception(
                "Параметр для вычисления не должен быть равным
нулю");
        return result;
    }
```

Предложенный метод обеспечивает деление, а так как Царевна и король никогда не делят на ноль, они все делят поровну, то с помощью оператора throw порождается исключение в случае неправильно введенных данных. Это исключение можно «отловить» с помощью catch по месту использования метода.

Жизнь царевны и короля изменилась: пришлось осваивать азы науки и знакомиться с простыми типами (чисел), разбираться в ветвлениях налогового кодекса, использовать все три типа циклов, а также усвоить исключения. Оставим их осмыслять все это до следующего номера... LXF

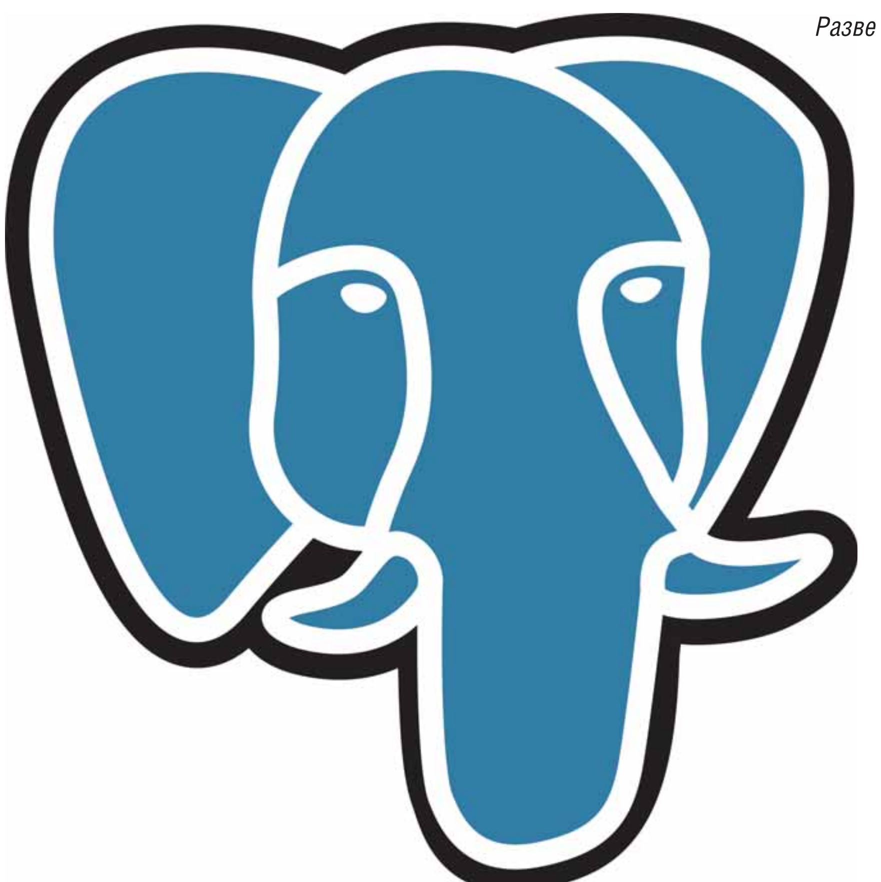


» **Через месяц** Мы научим Несмеяну работать с файлами и вести дневник.

Новая серия! Изучаем самый мощный из открытых серверов баз данных

PostgreSQL:

ЧАСТЬ 1: **Евгений Балдин** готов познакомить вас с замечательной СУБД *PostgreSQL* – а также людьми, которые ее разрабатывают.



Разве же так можно? Разве же такие вещи алгоритмизируешь?

Магнус Ф. Редькин об определениях счастья.
«Понедельник начинается в субботу»

fiodata

id	fio
1	Иванов И.П.
2	Балдин Е.М.

phonedata

id	number
1	555-32-23
2	555-41-37
2	(8)555932391

SELECT fiodata.id=phonedata.id

fio	number
Иванов И.П.	555-32-23
Балдин Е.М.	555-41-37
Балдин Е.М.	(8)555932391

Получение новой таблицы из уже имеющихся.

Новая информация добывается потом и кровью. Чтобы не утерять найденное, ее надо сохранить. А чтобы потом суметь найти необходимое, ее следует структурировать. *PostgreSQL* предназначен для постоянного[1] хранения структурированных данных[2].

Это кто такой?

PostgreSQL – это реляционная база данных. *PostgreSQL* – это программный продукт с открытым исходным кодом и свободной (в прямом смысле этого слова) лицензией. Собственно говоря, этим все сказано.

Реляционная база данных

Информация в реляционных базах данных хранится в виде обычных плоских двумерных таблиц. Доступ к данным в таблице можно получить по ее имени. В таблице есть именованные столбцы (column) и строки (row) – очень простая и понятная концепция. Пользователю предоставляется набор операторов, результатом действий которых так же являются таблицы. Эта особенность реляционной базы данных называется замкнутость. Это очень важное свойство, так как в результате любых действий порождаются объекты того же типа, что и объект, над которым совершались эти самый действия. Следствием замкнутости является возможность применять к результату все имеющиеся в наличии операторы. Иными словами, можно пользоваться вложенными выражениями[3].

Открытый исходный код

PostgreSQL распространяется под лицензией BSD. Почему не GPL? Ответ разработчиков можно перевести[4] примерно так: «*PostgreSQL* создавался в Беркли (Berkeley), как, собственно говоря, и лицензия BSD. Эта лицензия служила нам верой и правдой много лет. От добра – добра не ищут. Просьба не начинать опять «флеймить» по этому поводу.»

Генеалогия

Понятие реляционных баз данных было предложено в 70-ых годах прошлого века сотрудником фирмы IBM Эдгаром Ф. Коддом (Edgar F. Codd). В то время это была революция в сфере хранения данных. Головокружительный успех идей Кодда был связан еще и с тем, что он сумел воплотить математическую абстракцию под названием «реляционная алгебра» в жизнь. Многие ответы на практические вопросы были найдены теоретически с использованием математики.

С тех пор прошло более тридцати лет, и новой революции пока не предвидится. Двумерные таблицы еще долго будут основным методом структурирования информации в силу исключительной простоты решения.

Как и в случае с TCP/IP, практическое воплощение теории в жизнь началось с того, что DARPA (Defense Advanced Research Projects Agency) дало денег профессору. Профессор Майкл Стоунбрэйкер (Michael Stonebraker) написал реляционную базу данных POSTGRES, первый релиз которой был сделан в 1987 году. Профессор Стоунбрэйкер писал базу не с нуля. Его проект основывался на одной из самых первых

[1] Постоянность означает сохранность данных, даже если программа перестала работать.

[2] Хранить можно и не структурированные данные, но это уже моветон.

[3] Вложенные выражения – это многоуровневые выражения, причем использование имен реальных таблиц обязательно только на самом низком уровне. В остальных случаях в качестве объектов действия могут быть вычисляемые выражения.

[4] Очень вольный перевод.

введение



реляционных баз данных Ingres, к созданию которой приложил руку сам Кодд – ее имя частично присутствует в названии проекта (POSTGRES – после Ingres).

POSTGRES использовался как для реальной работы в качестве *СУБД*, так и для исследования теории реляционных баз данных в стенах университетов. В 1994 году два студента – Андрэ Ю (Andrew Yu) и Джолли Чен (Jolly Chen) – добавили движок SQL, который уже к этому моменту стал бесспорным промышленным стандартом для реляционных *СУБД*. Так появился Postgres95, который в 1996 году сменил имя на *PostgreSQL*. Имя больше не менялось, но активная разработка не прекращается не на миг. Последней версией (по состоянию на лето 2006 года) была 8.1.4. Подробнее об истории можно узнать в стандартной документации, поставляемой с программой, или на сайте **http://www.postgresql.org**.

Семейство *Ingres/PostgreSQL* породило множество коммерческих реализаций[5] систем управления баз данных, благо лицензия это позволяет.

А как оно работает?

На рисунке показана схема работы типичного приложения. Процессу POSTMASTER, который существует всегда[6], на серверную машину посылается запрос на подключение. Если запрос на подключение проходит проверку, то POSTMASTER создает свою копию. Все дальнейшие операции между базой данных и клиентом проводятся через эту копию POSTMASTER. На каждое соединение создается своя копия – это позволяет производить все действия с данными непосредственно на сервере.

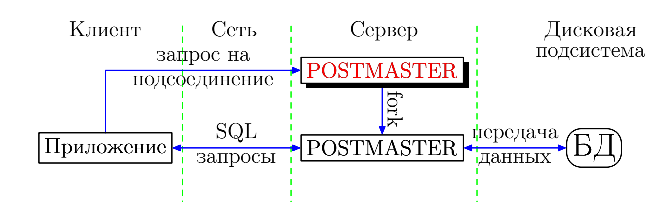


Схема работы приложения с PostgreSQL.

Установка и запуск

Установка базы данных – это не совсем тривиальная процедура. Лучше довериться стандартной сборке из вашего базового дистрибутива, даже если версия по умолчанию кажется устаревшей. Базовая версия *PostgreSQL* в Debian stable (Sarge) на момент написания статьи 7.4.7, в то время как последняя версия базы данных – 8.1.4. Различия есть, и весьма существенные, но для большинства задач вам хватит и 7.x. Самостоятельно собирать пакет из исходных текстов рекомендуется лишь в том случае, когда точно известно, что в базовой версии нет необходимой функциональности.

Если в будущем необходимо будет сменить версию *PostgreSQL*, то следует учитывать, что в случае крупных изменений (major releases) могут меняться внутренние форматы системных таблиц и файлов данных. В этих случаях необходимо выполнить процедуру dump/restore, которая гарантированно сохранит данные при «переезде». В отличие от крупных изменений, небольшие правки (minor releases[7]) не требуют никаких действий со стороны администратора БД.

Число пакетов, в описании которых упоминается *PostgeSQL*, довольно велико. Например, в Debian (Sarge) таких пакетов 182, что несколько меньше, чем число пакетов связанных с именем *mysql* (212), но превышает число упоминаний *InterBase/Firebird* (22), *sqlite* (50) и, естественно, Oracle (19). Это ни о чем не говорит, но корреляция, скорее всего, какая-то есть. К счастью, все 182 пакета ставить не обязательно – для Debian (Sarge) достаточно двух/трех:

```
# устанавливаются исполняемые файлы и файлы настроек
# необходимые для функционирования Базы Данных
> apt-get install postgresql
```

В случае подобной установки в обязательном порядке доставляется пакет *PostgreSQL*-client – базовый набор программ, которые можно ставить на клиентских машинах для удаленной связи с БД.

Если же, несмотря ни на что, хочется собрать все самостоятельно, то следует выполнить примерно следующую последовательность действий:

```
> wget ftp://ftp.postgresql.org/pub/source/v8.1.4/PostgreSQL-8.1.4.tar.bz2
> tar xvfj postgresql-8.1.4.tar.bz2
> cd postgresql-8.1.4
> ./configure
> make
> su
> make install
> adduser postgres
> mkdir /usr/local/pgsql/data
> chown postgres /usr/local/pgsql/data
> su - postgres
> /usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
> /usr/local/pgsql/bin/postmaster -D /usr/local/pgsql/data >logfile 2>&1 &
> /usr/local/pgsql/bin/createdb test
> /usr/local/pgsql/bin/psql test
```

Разберем то, что происходит, поподробнее. После того, как с помощью wget получен и распакован архив с исходными текстами, привычные команды **./configure** и **make** позволяют осуществить сборку *PostgreSQL*. Установку (*make install*) следует производить под суперпользователем (**su**). После установки необходимо добавить пользователя **postgres**, от имени которого и будет запущен сервер postmaster. По умолчанию установка программы производится в директорию **/usr/local/pgsql/**. Для хранения файлов базы предлагается создать директорию **/usr/local/pgsql/data**. Данные должны принадлежать пользователю **postgres** (команда chown). В этой же директории хранятся и файлы настройки.

Дальнейшая настройка производится под пользователем **postgres** (su – postgres). С помощью команды initdb производится инициализация хранилища данных, а вслед за этим производится запуск сервера postmaster. Последние две строчки создают тестовую базу данных test (createdb) и проверяют, что к ней можно подсоединиться (psql). Если все прошло нормально, то должно появиться приглашение вида:

```
Welcome to psql 8.1.4, the PostgreSQL interactive terminal.
```

[5] Особенно много потомков у *Ingres* — та же *Sybase*. Код *Sybase*, в свою очередь, в 1992 году был продан одной известной фирме, которая чуть позже выпустила продукт, в названии которого есть имя этой фирмы и слова «SQL Server». У POSTGRES в прямых потомках ходит Informix.

[6] За исключением тех случаев, когда компьютер выключен или сервис был остановлен администратором и очень редко по причине какой-либо ошибки. Я не знаю, какая статистика у других, но из моего опыта все ошибки такого рода были связаны с человеческим фактором.

[7] Меняется только последнее число в версии, то есть переход от версии 7.4.0 к версии 7.4.1.

```
Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

test=#
```

При установке стандартными средствами дистрибутива описанные выше действия выполняются автоматически, кроме последних двух строчек. В случае Debian (Sarge) при установке *PostgreSQL* можно указать, где именно расположить директорию с данными. По умолчанию все помещается в **/var/lib/postgres/data**. Для других дистрибутивов возможны вариации. Для выяснения подробностей следует изучить README. Например, в случае Debian, особенности пакета, связанные с дистрибутивом, описаны в **/usr/share/doc/postgresql/README.Debian.gz**. Ниже, если не указано специально, все действия выполняются для дистрибутива Debian (Sarge).

Для администрирования базы данных нет необходимости становиться root. Для этого можно настроить *sudo* (man sudo), то есть в файл **/etc/sudoers** (man sudoers) следует добавить примерно следующие строки:

```
# /etc/sudoers
Host_Alias HOME = localhost
User_Alias DBADM = "ваше имя, если Вы администратор базы
данных"
Cmnd_Alias DB  = /etc/init.d/postgresql
DBADM HOME =  NOPASSWD: DB
DBADM HOME = (postgres) NOPASSWD: ALL
```

Файлы настройки принадлежат пользователю postgres, поэтому для их изменения необходимо иметь возможность заходить под этим пользователем:

```
> sudo -u postgres bash
> whoami
postgres
```

либо добавить себя в группу postgres и разрешить этой группе редактировать конфигурационные файлы в директории **/etc/postgres** (**chgrp + chmod g+w**).

Скрипт **/etc/init.d/postgresql** позволяет управлять процессом postmaster

```
> sudo /etc/init.d/postgresql
Usage: /etc/init.d/postgresql {start|stop|autovac-start|autovac-
stop|restart|autovac-restart|reload|force-reload|status}
> sudo /etc/init.d/postgresql status
pg_ctl: postmaster is running (PID: 10868)
Command line was:
/usr/lib/postgresql/bin/postmaster '-D' '/home/postgres/data'
```

Этот же скрипт используется для автоматического запуска сервера при загрузке компьютера. От дистрибутива к дистрибутиву название инициализирующего скрипта может меняться.

После настройки сервера необходимо создать базу данных:

```
> sudo -u postgres createdb "имя БД"
CREATE DATABASE
```

и завести пользователя:

```
> sudo -u postgres createuser "имя пользователя"
Разрешить новому пользователю создавать базы? (y/n) n
Разрешить новому пользователю создавать пользователей? (y/n) n
CREATE USER
```

Для того чтобы пройти проверку при запросе на подключения необходимо, чтобы конфигурационный файл **pg_hba.conf** был настроен соответствующим образом. Например, чтобы можно было подключаться к базе данных под тем же именем, под которым вы работаете в системе, в **pg_hba.conf** должны присутствовать примерно следующие строки:

```
#/etc/PostgreSQL/pg_hba.conf
# TYPE  DATABASE  USER IP-ADDRESS   IP-MASK       METHOD
# connections by UNIX sockets
local   all       all                             ident sameuser
# All IPv4 connections from localhost
host    all       all  127.0.0.1 255.255.255.255 ident sameuser
```

Здесь в качестве метода идентификации используется метод ident sameuser[8]. Создав пользователя в соответствии с текущей учетной записью, можно подсоединиться к *PostgreSQL* и начать общаться с сервером базы данных на его родном языке – SQL:

```
> psql "имя БД"

"имя БД"=> SELECT fio,number FROM fiodata, phonedata
"имя БД"=>       WHERE fiodata.id=phonedata.id;
      fio      |       number
--------------+----------------
 Иванов И.П.  | 555-32-23
 Балдин Е.М.  | 555-41-37
 Балдин Е.М.  | (+7)5559323919
(записей: 3)
```

Почему?

Люди обычно работают с текстовыми файлами. Подавляющий объем структурированной информации до сих пор доставляется до нашего сознания через тест. Без помощи компьютера, в процессе ручного набора текста, информации производится не так уж и много. Необходимость базы данных в начале пути накопления личной информации не является очевидной. Все сделанное, в принципе, возможно окинуть взглядом.

Почему БД?

То, что создал один человек, другой человек, с высокой степенью вероятности, освоить в состоянии, но разобраться с наследием двух и более людей становится трудновато. А если это не наследие, а информация, идущая в реальном времени из многих (десятков, сотен, тысяч, миллионов) источников? Для начала все это надо куда-то сохранить, то есть необходимо надежное хранилище, по возможности ни от чего не зависящее.

И это еще полдела: данные надо как-то извлечь, причем не абсолютно все (иначе человеческий мозг в них утонет), а только нужные. Компьютеры пока еще не умеют надежно[9] считывать человеческие мысли, поэтому для начала необходимо нужные данные как-то пометить, и лучше это сделать в момент «укладки» в хранилище. То есть хранилище должно быть структурированным, причем структуру можно задавать извне до появления данных.

Когда данных немного – жить можно и так, оставляя ключевую информацию на обрывке листика, надеясь что он не затеряется. Обрывок листика слабо отличается от записи в каком-то файле. Текстовые утилиты типа *grep* существенно облегчают поиск информации, но всегда, в конце концов, настает момент, когда данных становится либо слишком много, либо они слишком часто изменяются и нужно вводить систему – *Систему Управления Базой Данных* или *СУБД*.

Почему PostgreSQL?

Когда я примерно шесть лет назад пытался понять, какую *СУБД* следует использовать для обеспечения эксперимента, в котором я участвую до сих пор, то выбора просто не существовало. Из свободных *СУБД* только *PostgreSQL* на тот момент обладал необходимой функциональностью. На сегодня вопрос выбора немного усложнился: подрос в хорошем смысле этого слова MySQL (в последней, 5-ой версии, говорят, наконец-то даже триггеры появились), были открыты исходные тексты проекта *Firebird*, в девичестве *Interbase* от фирмы Borland, да и «игрушечные» проекты типа *SQLite* тоже не лишены определенных преимуществ. Ну и, естественно, свет клином на открытых проектах не сошелся – тот

[8] Существует более либеральный метод проверки trust — в этом случае пускается кто угодно и под каким угодно пользователем. То есть метод «двери настежь» — некоторым нравится.

[9] Удачные опыты по управлению курсором мыши или манипулятором уже зафиксированы, правда, для этого требуются имплантанты. Без имплантантов операторы могут передавать только самые простейшие команды, и вряд ли в ближайшее время ситуация кардинально изменится.

К вопросу о номере порта

По умолчанию для создания TCP/IP соединения *postmaster* использует порт номер 5432. Если номер порта отличается от установленного по умолчанию, то *postmaster* должен быть запущен с ключом **-p** [номер порта]. Для выяснения этого достаточно выполнить:

```
> ps axw | grep
postmaster | grep -v grep
 4181 ?     S     0:00 /
usr/lib/postgresql/bin/
postmaster -D /home/
postgres/data
```

Номер порта может также храниться в переменной окружения $PGPORT.

же Oracle [бесплатно] предлагает свои *СУБД* для изучения. И все-таки я выбираю *PostgreSQL* – решение шестилетней давности меня не разочаровало. На редкость устойчивая к внешним воздействиям программа с абсолютно предсказуемым поведением. Даже те случаи, которые мне по неопытности показались «граблями», оказались «фичами».

Одной из основных целей, которая была поставлена при разработке *PostgreSQL*, является строгое соответствие стандартам. *PostgreSQL* поддерживает ANSI SQL–92, SQL–99 (SQL–2 и SQL–3, соответственно), а также многие из возможностей ANSI SQL:2003. Мало кому[10] удается похвастаться подобным соответствием стандартам.

В дополнение к стандартам, *PostgreSQL* поддерживает множество полезных расширений. Примером мелкого, но полезного расширения, не входящего в стандарт SQL, являются дополнения к условию для SELECT вида LIMIT/OFFSET[11], которые позволяют получить только указанные строки из результата запроса. *PostgreSQL* полностью поддерживает механизм транзакций (transactions), вложенные запросы (subselects), триггеры (triggers), представления (views), функциональные индексы, ссылочную целостность по внешнему ключу (foreign key referential integrity), изощренные типы блокировок (sophisticated locking) и многое другое.

К названию *PostgreSQL* обычно прибавляется слово «объектная», то есть полное наименование звучит как «объектно-реляционная база данных *PostgreSQL*». Пользователю предоставляются необходимые инструменты для создания новых типов данных, функций, операторов и своих методов индексирования. Подобные возможности позволяют работать с довольно нестандартными данными, например, с картографическими объектами – PostGIS (**http://postgis.refractions.net/**).

Размер базы данных, управляемой *PostgreSQL,* не ограничен, также нет ограничения и на число строк в таблице. Да и вообще, есть ли ограничения у этого чуда? Да, есть: ваша таблица не может быть больше чем 32 Тбайта, а число столбцов в таблице не может быть больше 250–1600, в зависимости от типа данных. Много это или мало? Зависит от задачи: я, например, как-то уперся в ограничение по числу столбцов, но скорее по неопытности, нежели по необходимости. Описанное выше верно для версии *PostgreSQL* 8.1.4. Возможно, в будущем будут сняты и эти ограничения.

Существует родные интерфейсы для работы с *PostgreSQL* из языков Java (JDBC), Perl, Python, Ruby, C, C++, PHP, Lisp, Scheme и всего, что может связаться через ODBC. *PostgreSQL* поддерживает хранимые процедуры, которые можно написать на множестве языков программирования, включая Java, Perl, Python, Ruby, Tcl, C/C++ и родном для *PostgreSQL* PL/pgSQL.

По результатам автоматизированного тестирования, проведенном в 2005 году (**http://www.postgresql.org/about/news.363**), в коде *PostgreSQL* было обнаружено 20 дефектов, что соответствует 1 ошибке на 39 тысяч строк кода. Для сравнения, аналогичное тестирование примерно в то же время выявило по одному дефекту на 10 тысяч строк кода в ядре Linux, а в MySQL одно проблемное место приходится на 4 тысячи строк кода. Это ни о чем не говорит, так сказать, мелочь, зато разработчикам и пользователям *PostgreSQL* приятно.

Информация о...

Книг по *PostgreSQL*, выпущенных на русском языке, относительно[12] немного, но они есть, и количество их будет расти. Эта область технических знаний не так популярна, как следовало бы. Очевидно, что в будущем без надежных хранилищ данных будет непросто управляться со все возрастающим потоком информации.

С другой стороны, наличие отличной документации (в том числе и русскоязычной) позволяет достаточно безболезненно «войти в тему». Вполне можно обойтись и без специфичных для *PostgreSQL* возможностей, а для изучения основ SQL годится любая нормальная книга, коих довольно много. Для введения вполне сгодится «SQL» от Мартина »

[10] Возможно, что вообще никому. В большинстве случаев, следование стандарту заканчивается на вводном (entry) уровне SQL-92.

[11] Мне эти инструкции в свое время сильно облегчили жизнь, точнее, увеличили скорость выполнения нужных мне запросов.

PostgreSQL в лицах: Олег Сергеевич Бартунов

Среди разработчиков PostgreSQL можно встретить немало наших соотечественников. Мы в Linux Format решили, что, рассуждая о преимуществах этой открытой СУБД, будет несправедливо не рассказать вам об этих людях, тем более, что мы говорим с ними — в прямом и переносном смысле — на одном языке.



Олег Бартунов добавил в *PostgreSQL* поддержку локали, а сейчас активно работает над улучшением алгоритмов индексирования и поиска. В фокусе его текущей деятельности в команде разработчиков *PostgreSQL* находится структура GiST (обобщенное поисковое дерево) и множество приложений с ее использованием.

Евгений М. Балдин (ЕМБ): Меня интересуют причины, по которым вы включились в проект *PostgreSQL*. Не проще ли было подождать, пока кто-то другой сделает поддержку русской локали?

Олег С. Бартунов (ОСБ): Я – астроном, а астрономия использует базы данных практически везде. Будучи в UCSC (Univerity California Santa Cruz), я познакомился с Postgres, который разрабатывался недалеко в UC Berkeley. Позже, в 1996 году, я работал над сайтом «Учительской газеты», и мне надо было сделать поисковую систему по архиву. Я взял Postgres95, разобрался, как его скомпилировать под Linux, что было не так просто, и загрузил данные. Я был неприятно удивлен, что у меня ничего не работает. Американцев проблема локали не интересовала, а европейских пользователей было немного, тем более в России. Так что ждать помощи было не от кого. На тот момент я уже имел опыт работы с локалью: в 1995 году принимал участие в обсуждении и тестировании локали в Perl.

ЕМБ: Раньше было гораздо проще с выбором - выбирать было не из чего. А сейчас, с текущим разнообразием свободных и доступных СУБД, вы все равно бы выбрали *PostgreSQL*?

ОСБ: Да, *PostgreSQL* рулит. Мне нравится его идеология и сильная команда разработчиков.

ЕМБ: Изучая вашу домашнюю страничку, я наткнулся на упоминание о книге под названием «*Рецепты PostgreSQL*». Это в планах? Когда будет?

ОСБ: Действительно, меня и Алексея Борзова на это подбивает издательство БХВ (http://www.bhv.ru/). В подобной книге есть необходимость, но – никак не получается. Мы не профессиональные писатели, а на написание хорошей книги требуется много усилий и времени. Я думаю, что без спонсорства написание книги может занять очень много времени.

ЕМБ: Ну и в заключение: судя по набору поддерживаемых вами проектов, вы занимаетесь работой по своей специальности. Участие в техническом проекте уровня *PostgreSQL* не отвлекает от основной работы?

ОСБ: Все мои проекты в той или иной степени используют *PostgreSQL*, нашу работу поддерживает РФФИ (http://www.rfbr.ru/). Сейчас мы активно работаем по теме «Виртуальная Обсерватория». Центр находится у нас в ГАИШ, и многотерабайтное хранилище там построено именно на *PostgreSQL* (http://vo.astronet.ru/).

Азбука SQL: А



Язык определения данных

Группа операторов SQL, ответственных за описание и модификацию структуры БД, представляет собой специализированный язык определения данных (Data Definition Language).

Язык определения данных включает следующие операторы:

- **CREATE** – позволяет создать новую базу данных, таблицу (TABLE), индекс (INDEX), представление (VIEW) или какую-либо другую сущность,
- **DROP** – позволяет удалить то, что было создано с помощью CREATE,
- **ALTER** – позволяет изменить параметры уже существующего объекта.

Из перечисленных наиболее популярна команда создания таблиц:

```
CREATE TABLE «имя таблицы» (
  «имя столбца 1» «тип данных»
    [DEFAULT «значение по умолчанию»] [«ограничения на столбец»],
  [«имя столбца 2» ...,]
  [...,]
  [«ограничения на таблицу»]
)
```

Ключевое слово **DEFAULT** предваряет значение, которое присваивается, если при вставке данных отсутствует инициализация.

Ограничения, накладываемые на столбец, могут характеризоваться ключевыми словами:

- **NOT NULL** – инициализация при вставке обязательна.
- **UNIQUE** – значение должно быть уникальным.
- **PRIMARY KEY** – значение объявляется первичным ключом. В таблице может быть ровно один первичный ключ. Значение, объявленное первичным ключом должно быть определенным и уникальным (**NOT NULL+UNIQUE**)
- **CHECK(«условное выражение»)** – проверка значения.
- **REFERENCES** – ссылка на допустимый диапазон значений, представленных во внешней таблице (внешний ключ **FOREIGN KEY**).

Ограничения, накладываемые на таблицу в целом, дублируют описанную функциональность, позволяя при составлении требований использовать имена несколько столбцов. Например, первичный ключ вполне может быть составным.

DROP TABLE позволяет уничтожить таблицу, созданную **CREATE TABLE**. Синтаксис команды уничтожения гораздо проще команды созидания. Надо только добавить «имя таблицы». Это очень опасная для данных команда.

Довольно редко, но все же бывает необходимо изменить уже существующую таблицу:

```
ALTER TABLE «имя таблицы» «действие»
```

В качестве «действия» можно добавить столбец или ограничение (**ADD**), удалить столбец или ограничение (**DROP**), изменить параметра столбца (**ALTER**), а так же «повесить»/«выключить» триггер (**ENABLE/DISABLE TRIGGER**).

Грабера (издательство «Лори», 2003). Собственно говоря, хватит и стандартной документации, которая идет в дистрибутиве.

Основной сайт *PostgreSQL* находится по адресу: http://www.postgresql.org. Там расположено первичное хранилище обширной документации, в которой есть фактически вся «мудрость мира», имеющая хоть какое-то отношение к *PostgreSQL* – надо только уметь читать.

По адресу http://www.linuxshare.ru/postgresql/ представлена русскоязычная версия сайта. Там же можно найти информацию о русскоязычном тематическом списке рассылки pgsql-ru-general-owner@postgresql.org. Список не сильно активный, но если хочется перемолвиться о «subj» по-русски – вполне сгодится. LXF

[12] Например, относительно числа книг по PHP+MySQL.

 **Через месяц** Мы приступим к непосредственной работе с базой данных.

Набор математики:

ЧАСТЬ 3: Успешно пройдя две ступени посвящения, вы готовы двинуться в путь к вершинам мастерства. Вашим проводником в этом нелегком деле будет **Евгений Балдин**.



Полиграфисты относят
математические работы к каторжным...
Д.Э. Кнут. Математическая типография.

Иногда от незнакомых с ТеХнологиями людей приходится слышать, что *LaTeX* годится только для набора математики. При знакомстве же с истинными ТеХнологиями возникает понимание, что *LaTeX* настолько хорош, что с его помощью можно набирать даже математику.

Набор математических формул всегда считался вершиной типографского искусства. Дело в том, что формулы, для концентрации информации и дополнительной выразительности по сравнению с обычным текстом, являются многоуровневыми. Д.Э. Кнут к своей программе компьютерной типографии создал и язык для описания формул. После короткого периода обучения пользователь в состоянии читать и набирать на этом языке формулы практически любой сложности.



LaTeX – не единственная программная среда, использующая *TeX*-нотацию. Эта же нотация рекомендуется при наборе всех сколько-нибудь сложных формул на страницах Википедии (http://ru.wikipedia.org статья «Википедия:Формулы»).

Становлению *TeX* как стандарта для набора формул в значительной степени поспособствовало Американское математическое сообщество (The American Mathematical Society – AMS), которое субсидировало разработку расширения *TeX*, известного как *AMSTeX*, в начале восьмидесятых годов прошлого столетия. В 1987 году наработки AMSTeX были добавлены в *LaTeX* в виде пакета *amsmath*. Вместе с *amsmath* в *LaTeX* было добавлено множество улучшений, позволяющих набирать действительно изощрённую математику. Поэтому при использовании в тексте математики в шапке документа следует в обязательном порядке загружать пакет *amsmath*:

```
\usepackage{amsmath}
```

В дальнейшем предполагается, что этот пакет уже загружен.

Полностью описать все команды языка для набора формул в рамках короткой статьи нереально, так как математика, как и способы её представления, безгранична. Поэтому основное внимание будет уделено базовым правилам и русскому стилю в формулах. В любой сколько-нибудь большой книге по *LaTeX* будет полный список всех команд. Если серьёзно работать с математикой, то подобная книжка, в любом случае, понадобится.

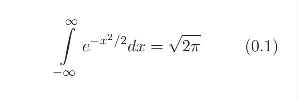
Набор формул

При формировании текста формулы подразделяются на строчные и выносные. Строчные формулы набираются внутри абзаца вместе текстом. По описанию формулы *LaTeX* создаёт бокс, который обрабатывается наравне с обычными текстовыми боксами. Как правило, строковые формулы это небольшие вставки, вроде $E=mc^2$. Выносные или выключенные формулы выводятся за пределы абзаца.

Строчная формула в тексте ограничивается[1] с помощью символа доллара $«формула»$ или с помощью команд-скобок \(«формула»\). При наборе предпочтительнее использовать второй вариант оформления, так как он позволяет легко определить, где начинается, а где кончается формула. «Долларовое» ($) окружение лучше тем, что оно чуть-чуть короче, кроме этого, команда $ крепкая[2], в отличие от команд-скобок.

Однострочные выносные формулы формируются с помощью окружения equation. Так как в этом случае формула вынесена за пределы абзаца, то её можно пронумеровать. Например:

```
\begin{equation}
  \label{eq:math:ex1}
  \int\limits_{-\infty}^{\infty}
          e^{-x^2/2}dx=\sqrt{2\pi}
\end{equation}
```



$$\int\limits_{-\infty}^{\infty} e^{-x^2/2}dx=\sqrt{2\pi} \qquad (0.1)$$

[1] Есть более формальное оформление строчной формулы как окружения: \begin{math} «формула» \end{math}. Но, в силу понятных причин, никто подобное описание не использует.

[2] Начав изучать команды *LaTeX*, довольно быстро сталкиваешься с понятиями «хрупкости»/«крепкости». «Крепкие» команды, в отличие от «хрупких», можно использовать в качестве аргументов других команд. С другой стороны, хрупкие команды тоже можно использовать как параметры, защитив их с помощью команды \protect. Эти понятия в большинстве своём пережитки прошлого и их постепенно изживают, но пока следует иметь их в виду.

Месяц назад Мы освоились с базовыми элементами LaTeX.

Снова в школу

Нумерация формул удобна для того, чтобы позже в тексте на неё можно было легко сослаться с помощью команды \eqref{eq:math:1}[3]. Если же формул немного и не хочется никакой нумерации, то можно воспользоваться окружением equation*[4].

При создании выключенной формулы размер шрифта для улучшения читаемости немного увеличивается. *LaTeX* имеет несколько стилей для оформления математических формул. При желании можно выбрать необходимый стиль вручную:

- \displaystyle – стиль, используемый для выносных формул,
- \textstyle – стиль строчных формул,
- \scriptstyle – в этом стиле набираются индексы,
- \scriptscriptstyle – индексы второго уровня.

С помощью этих команд можно увеличить размер шрифта для формул внутри абзаца, или заставить индексы выглядеть как базовые символы. Например, сравните:

```
\begin{equation*}
  \frac{1}{1+
    \frac{1}{1+
      \frac{1}{1+
        \frac{1}{2}}}}
\end{equation*}
```

$$\frac{1}{1+\frac{1}{1+\frac{1}{1+\frac{1}{2}}}}$$

и

```
\begin{equation*}
  \frac{1}{\displaystyle 1+
    \frac{1}{\displaystyle 1+
      \frac{1}{\displaystyle 1+
        \frac{\displaystyle 1}
             {\displaystyle 2}}}}
\end{equation*}
```

$$\frac{1}{\displaystyle 1+\frac{1}{\displaystyle 1+\frac{1}{\displaystyle 1+\frac{\displaystyle 1}{\displaystyle 2}}}}$$

Пробелы в формулах отмечают только конец команды, а сами по себе смысла не имеют – *LaTeX*, как правило, гораздо лучше знает, как сформировать результат.

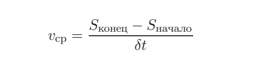
Кириллица в формулах

Всё дело в имеющихся шрифтах – они красивые, разнообразные, но в большинстве своём англоязычные. В настоящее время кириллические математические шрифты в «дикой природе» отсутствуют, поэтому приходится пользоваться их текстовыми версиями.

Стиль mathtext (пакет *t2*), позволяет использовать кириллицу в формулах без дополнительных ухищрений. Стиль может быть подключён с опцией warn – в этом случае он сообщает обо всех случаях использования кириллических букв в формулах. *mathtext* следует загружать до *babel* и/или *fontenc*.

```
\usepackage[warn]{mathtext}
```

```
\[
 v_{cp}=\frac{S_{конец}-S_{начало}}
             {\delta t}
\]
```



Здесь для создания выключенной формулы используется команда \[«формула»\] – краткий аналог окружения equation*. В отличие от латиницы, русские буквы в формулах печатаются прямым шрифтом – это было сделано специально. Чтобы изменить это поведение, в преамбуле следует добавить команду для переопределения шрифта:

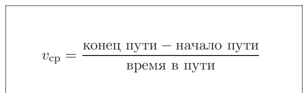
```
\DeclareSymbolFont{T2Aletters}{T2A}{cmr}{m}{it}
```

[3] Метка выставляется с помощью команды \label.

[4] К equation добавляется звёздочка. Подобный приём в создании команд применяется достаточно часто. Команда со * обычно не нумеруется и не отображается ни в каких автоматически составляемых списках.

Стиль amstext (загружается автоматически при загрузке *amsmath*) определяет команду \text, которая позволяет вставлять в формулу обычный текст. Он может быть и русским:

```
\[v_{cp}=
   \frac{\text{конец пути}-
         \text{начало пути}}
        {\text{время в пути}}\]
```



Преимущество такого подхода заключается в том, что внутри команды \text пробелы воспринимаются как нормальные символы и слова не сливаются. Использование \text предпочтительно и для целей переносимости.

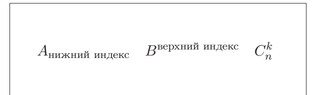
Школьная математика

Математика в школе – это явление, через которое проходит каждый. Именно поэтому фактически любой вменяемый россиянин умеет обращаться с дробями, знает теорему Пифагора, с лёгкостью решает квадратные уравнения и что-то слышал про интеграл и производную. Разберёмся со всем этим поподробнее.

Индексы

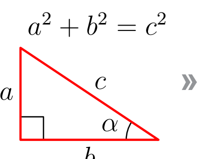
Букв в латинском алфавите не так уж и много, а научных понятий – без числа. Один из способов отличать обозначения друг от друга – это индексы, как верхние, так и нижние:

```
\[A_{\text{нижний индекс}}\quad
  B^{\text{верхний индекс}}\quad
  C_n^k\]
```



Обратите внимание, что если в индексе ровно один знак, то фигурные скобки вокруг него можно и нужно опустить. Теперь мы можем записать теорему Пифагора: \(a^2+b^2=c^2\)

$$a^2+b^2=c^2$$



»

Математические символы

Кроме символов латиницы и кириллицы, математики используют множество самых разнообразных значков, да и латиница не так уж проста. Если воспользоваться пакетом *amsfonts*, то она может стать такой:

```
\begin{itemize}
 \item \(ABCD\) "--- обычной,
 \item \(\mathbf{ABCD}\) "--- жирной,
 \item \(\mathbb{ABCD}\) "--- ажурной,
 \item \(\mathcal{ABCD}\) "---
                           прописной.
\end{itemize}
```

- $ABCD$ — обычной,
- $\mathbf{ABCD}$ — жирной,
- $\mathbb{ABCD}$ — ажурной,
- $\mathcal{ABCD}$ — прописной.

Это далеко не все возможные шрифтовые стили которые можно применять в математической моде, но лучше особо не перегружать формулы всякой «готикой» (например, \mathfrak).

Не единой латиницей жив математик. Традиционно, везде, где только можно, используются греческие буквы.

Греческие символы									
α	\alpha	β	\beta	γ	\gamma	δ	\delta	ε	\epsilon
ζ	\zeta	η	\eta	θ	\theta	ι	\iota	$\varkappa$	\kappa
λ	\lambda	μ	\mu	ν	\nu	ξ	\xi	o	o
π	\pi	ρ	\rho	σ	\sigma	τ	\tau	υ	\upsilon
φ	\phi	χ	\chi	ψ	\psi	ω	\omega	Γ	\Gamma
Δ	\Delta	Θ	\Theta	Λ	\Lambda	Ξ	\Xi	Π	\Pi
Σ	\Sigma	Υ	\Upsilon	Φ	\Phi	Ψ	\Psi	Ω	\Omega

» В *LaTeX* присутствует полный набор и, за исключением трёх букв, начертание вполне привычное. Для исправления непривычных начертаний эти буквы были переопределены с помощью пакета *amssymb*:

```
%Переопределение kappa epsilon phi на русский лад
\renewcommand{\kappa}{\varkappa}
\renewcommand{\epsilon}{\varepsilon}
\renewcommand{\phi}{\varphi}
```

Спецсимволов в *LaTeX* великое множество. В стандартной поставке *TeX Live* идёт «Всеобъемлющий список сиволов LaTeX» (The Comprehensive LaTeX Symbols List – файл **symbols-a4.pdf**) в котором перечислено 3300 распространённых символов, применяемых пользователями *LaTeX*. Почти наверняка любой операнд, который вам нужен, там уже есть. Ниже будет перечислена только та часть символов, которая, с моей точки зрения, может пригодиться в наборе школьной математики. Пакет *amssymb* для использования обязателен.

«Школьные» символы									
$\hat{a}$	\hat{a}	$\bar{a}$	\bar{a}	$\vec{a}$	\vec{a}	$\dot{a}$	\dot{a}	$\tilde{a}$	\tilde{a}
$\pm$	\pm	$\mp$	\mp	$\times$	\times	$\cdot$	\cdot	$\div$	\div
$\vee$	\lor	$\wedge$	\land	$\neg$	\neg	$\forall$	\forall	$\exists$	\exists
$\leqslant$	\le	$\geqslant$	\ge	$\ll$	\ll	$\gg$	\gg	$\neq$	\neq
$\equiv$	\equiv	$\sim$	\sim	$\simeq$	\simeq	$\approx$	\approx	$\propto$	\propto
$\parallel$	\parallel	$\perp$	\perp	$\angle$	\angle	$\triangle$	\triangle	$\sphericalangle$	\sphericalangle
∞	\infty	ℓ	\ell	$\sum$	\sum	$\prod$	\prod	$\varnothing$	\varnothing

Для соответствия русским традициям 2 символа были переопределены:

```
%Переопределение le ge на русский лад
\renewcommand{\le}{\leqslant}
\renewcommand{\ge}{\geqslant}
```

Дроби

Дроби формируются с помощью команды \frac[5]:

```
\[
 дробь=\frac{числитель}{знаменатель}
\]
```

$$\text{дробь} = \frac{\text{числитель}}{\text{знаменатель}}$$

Как и практически вся математика в *LaTeX*, дробь записывается так, как читается само выражение.

Корни

Для рисования знака корня используется команда

```
\sqrt[степень]{«подкоренное выражение»}
```

Степень можно упустить. В этом случае рисуется обычный квадратный корень.

```
\[
 \overline{
  \underline{\Large
   \sqrt[3]{a}+\sqrt[2]{b}+\sqrt[99]{g}
  }
 }
\]
```

$$\overline{\underline{\sqrt[3]{a}+\sqrt[2]{b}+\sqrt[99]{g}}}$$

Обратите внимание, что знак корня размещается в соответствии с размерами подкоренного выражения. Если в выражении присутствует только один корень, то это самое разумное поведение, но в случае нескольких корней, как в приведённом выше примере, не помешает выравнивание.

Для выравнивания по высоте используется команда \mathstrut[6]. В результате её применения вставляется невидимый символ нулевой толщины и высоты, в точности равной высоте круглой скобки:

```
\[\Large
  \sqrt[3]{\mathstrut a}+
  \sqrt[2]{\mathstrut b}+
  \sqrt[99]{\mathstrut g}
\]
```

$$\sqrt[3]{\mathstrut a}+\sqrt[2]{\mathstrut b}+\sqrt[99]{\mathstrut g}$$

Квадратное уравнение

И, наконец, вершина школьной математики – это решение квадратного уравнения $ax^2+bx+c=0$:

```
\[
 x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\]
```

$$x_{1,2}=\frac{-b\pm\sqrt{b^2-4ac}}{2a}$$

Теперь можно смело садиться за написание методичек.

Функции

Все символы в математической моде печатаются курсивом, поэтому названия функций, для выделения, печатаются прямым шрифтом. Кроме смены шрифта, функции с обоих сторон должны правильно «отбиваться» пробелами, иначе будет некрасиво. При загрузке русского языка с помощью пакета *babel* кроме стандартных имён функций доопределяется несколько сокращений, применяемых в русскоязычной литературе. Среди часто употребляемых функций можно упомянуть: cos, arccos, sin, arcsin, tg, arctg, ctg, arcctg, sh, ch, th, cth, exp, ln, log, lim, min и max. В математической моде эти функции можно использовать в качестве команд:

```
\begin{equation*}
 \begin{split}
  &\log_2 10=\ln10/\ln2\simeq3.32 \\
  &\lim_{x\to0}\frac{\sin x}{x}=1\\
  &(a+b)^n=\sum_{k=1}^n C^k_n a^kb^{n-k}
 \end{split}
\end{equation*}
```

$$\log_2 10=\ln10/\ln2\simeq3.32$$
$$\lim_{x\to0}\frac{\sin x}{x}=1$$
$$(a+b)^n=\sum_{k=1}^n C^k_n a^kb^{n-k}$$

Обратите внимание на обработку индексов для функции log (логарифм) и lim (предел). Для доопределения новых функций правильнее всего воспользоваться в преамбуле командой DeclareMathOperator:

```
%В преамбуле - определение новых функций
\DeclareMathOperator{\log-like}{log-like}
\DeclareMathOperator*{\lim-like}{lim-like}
```

В зависимости от варианта команды индексы отображаются как для логарифма (команда без звёздочки) или как для предела (команда со звёздочкой).

Производная и интеграл

В старших классах, в конце обучения, обычно чуть-чуть касаются понятий интегрирования и дифференцирования. Возможно, для того, чтобы правильно подсчитать сдачу в магазине, эти знания не являются необходимыми. Но для изучения физики и, как следствие, химии и биологии без интегралов никак – поверьте мне на слово.

Производная обычно отмечается штрихом. В физике, производная по времени выделяется точкой, для того чтобы отличать её от производной по координате. Можно честно написать \frac{d F(x)}{dx}. Для частной производной вместо буквы d используется спецсимвол \partial:

```
\[ f'\quad f''\quad
  \dot{f}\quad \ddot{f} \quad
  \frac{d f}{d x}\quad
  \frac{\partial f}{\partial x} \]
```

$$f'\quad f''\quad \dot{f}\quad \ddot{f} \quad \frac{d f}{d x}\quad \frac{\partial f}{\partial x}$$

Взятие производной есть обратная операция по отношению к интегрированию:

```
\[
 \frac{d}{dx}\int F(x) dx=F(x)
\]
```

$$\frac{d}{dx}\int F(x) dx=F(x)$$

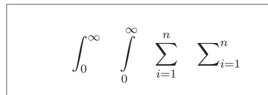
[5] От слова *fraction* – «дробь».

[6] От английского *strut* – «подпорка» или «страта».

Приглядевшись к этому примеру, можно отметить, что, вопреки русским математическим традициям, представленный здесь интеграл не прямой, а наклонный. Это можно исправить, например, загрузив пакет *wasysym* с опцией integrals. К сожалению, получающиеся интегралы «не смотрятся», поэтому пока лучше использовать начертания по умолчанию в надежде, что в будущем ситуация изменится к лучшему.

Неопределённый интеграл – это хорошо, а определённый – лучше. Качественное оформление пределов интегрирования важно для восприятия формулы.

```
\[
 \int_0^{\infty}\quad
 \int\limits_0^{\infty}\quad
 \sum_{i=1}^n\quad
 \sum\nolimits_{i=1}^n\quad
\]
```

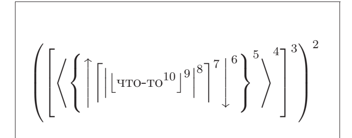


По умолчанию пределы размещаются справа от интеграла. Ситуацию можно поправить с помощью команды \limits. Команда \nolimits делает всё ровно наоборот.

Скобки

Для визуальной группировки символов внутри формулы скобки – вещь незаменимая. Особенно здорово, если скобки автоматически подгоняют свой размер под выражение, которое они окружают. Парные команды \left и \right включают режим подобной подстройки:

```
\[\left(
   \left[
    \left\langle
     \left\{
      \left\uparrow
       \left\lceil
        \left|
         \left\lfloor
         \text{что-то}^{10}
         \right\rfloor^9
        \right|^8
       \right\rceil^7
      \right\downarrow^6
     \right\}^5
    \right\rangle^4
   \right]^3
  \right)^2\]
```



Эстеты, в зависимости от ситуации, могут добавить в конце команды модификатор позиционирования разделителя как левого – l (отбивка как для \left), правого – r (отбивка как для \right) и среднего – m.

Перенос формул

В русскоязычной литературе принято, что при переносе строчной формулы на другую строку знак, по которому разрывается формула, дублируется на следующей строке. Например:

```
a + b =
= c
```

Однако, по умолчанию этого не происходит. Проще всего решить проблему с помощью следующего макроса[7], который необходимо определить в преамбуле:

```
%перенос формул в тексте
\newcommand*{\hm}[1]{#1\nobreak\discretionary{}%
 {\hbox{$\mathsurround=0pt #1$}}{}}
```

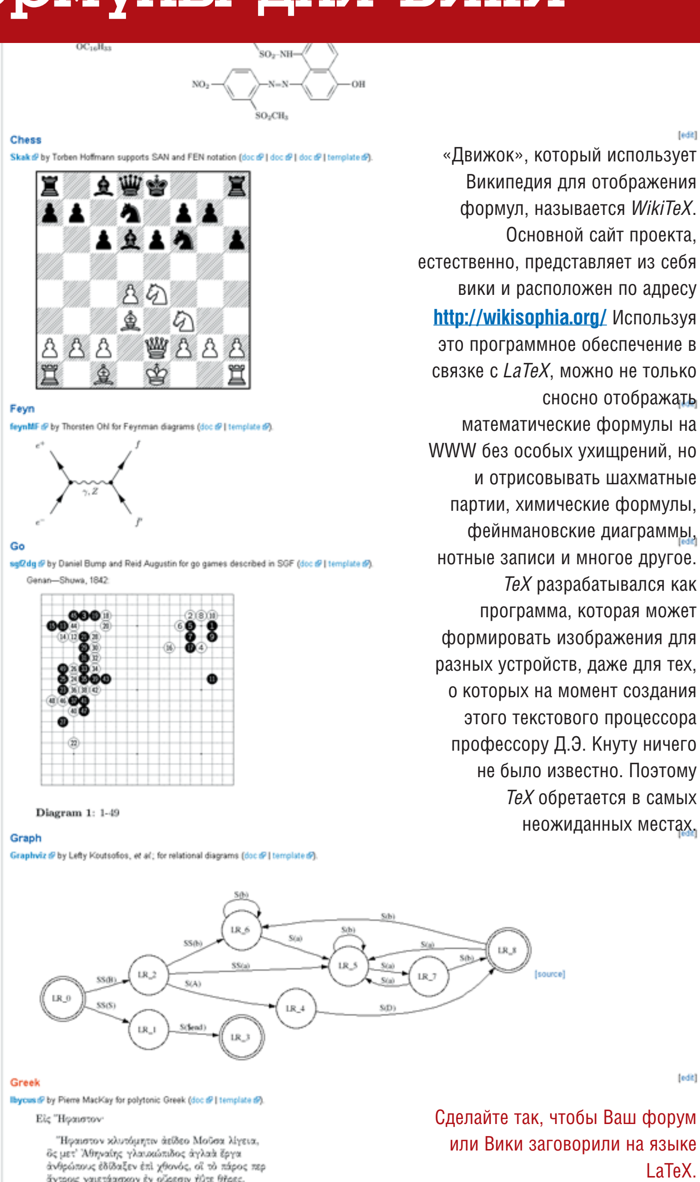
Здесь определена команда \hm, которую следует добавлять в местах потенциального переноса формулы, примерно, так: \(a + b \hm{=} c\). Сделать это можно во время окончательной доводки текста. В любом случае, для полировки рукописи ручная работа необходима.

Заключение

Изложенных правил и приёмов вполне хватит для набора текстов в рамках школьной математики. Для более изощрённых формул требуются более продвинутые приёмы и конструкции. Всё это будет, но чуть позже.

[7] Рецепт от Евгения Миньковского из fido7.ru.tex

Формулы для Вики



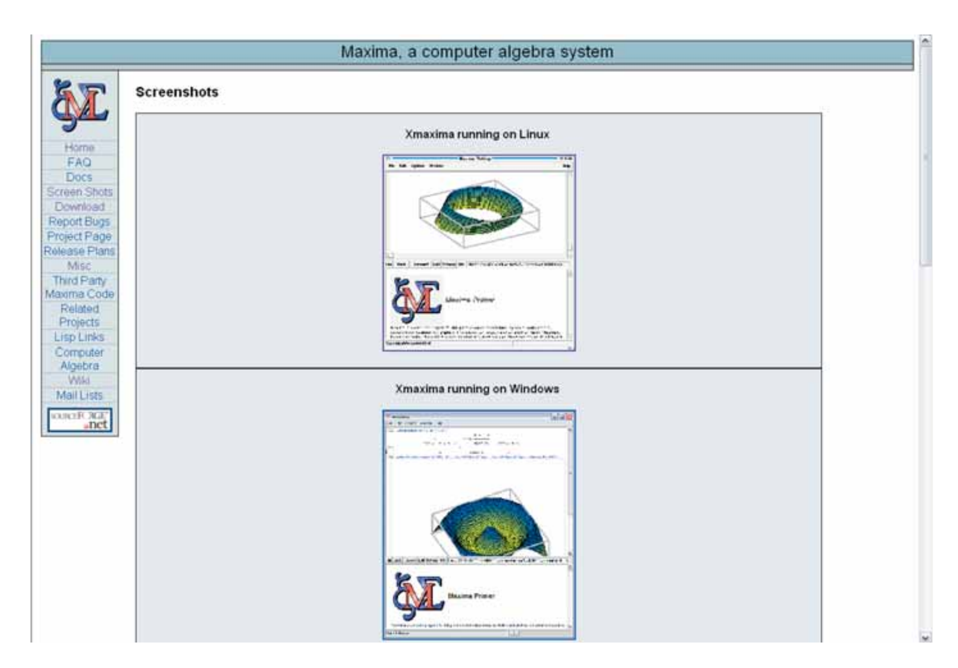
«Движок», который использует Википедия для отображения формул, называется *WikiTeX*. Основной сайт проекта, естественно, представляет из себя вики и расположен по адресу http://wikisophia.org/ Используя это программное обеспечение в связке с *LaTeX*, можно не только сносно отображать математические формулы на WWW без особых ухищрений, но и отрисовывать шахматные партии, химические формулы, фейнмановские диаграммы, нотные записи и многое другое. *TeX* разрабатывался как программа, которая может формировать изображения для разных устройств, даже для тех, о которых на момент создания этого текстового процессора профессору Д.Э. Кнуту ничего не было известно. Поэтому *TeX* обретается в самых неожиданных местах.

Сделайте так, чтобы Ваш форум или Вики заговорили на языке LaTeX.

Через месяц Твердо усвоив школьную программу, мы двинемся в сторону высшей математики.

Графики и управляющие

ЧАСТЬ 5 Сегодня **Тихон Тарнавский** продемонстрирует вам «графические» возможности Maxima, а также безупречную работу конструкций if, for и других...



«А рисовать вы тоже умеете?» – «Рисовать? Кого-нибудь привлечем.»

Как мы уже говорили в прошлый раз, количество различных функций в *Maxima* разработчики постарались свести к минимуму, а широту размаха каждой конкретной функции, соответственно, к максимуму. Соблюдается эта тенденция и в функциях построения графиков: основных таких функций всего две, с очевидными, как всегда, названиями – plot2d и plot3d (одно из значений слова «plot» – «график», а аббревиатуры 2d и 3d переводятся как «двумерный» и «трехмерный»). Если говорить точнее, возможности графической отрисовки не встроены в *Maxima*, а реализованы посредством внешних программ, в чем и прослеживается пресловутый Unix-way: «одна задача – одна программа». По умолчанию, построением графиков занимается *gnuplot*, но кроме него есть разрабатываемый вместе с *Maxima* и идущий в ее же пакете *openmath*. *Gnuplot* необходимо установить (вручную либо автоматически – как зависимость *Maxima*) из пакета *gnuplot-nox*, либо просто *gnuplot*, а для работы openmath нужен командный интерпретатор *wish*, входящий обычно в пакет *tk*; и, начиная с версии 5.10.0, еще и *xMaxima*.

Теперь кратко – о возможностях. Начнем с plot2d. Кратчайший вариант ее вызова такой: plot2d(выражение, [символ, начало, конец]), где выражение задает функцию, график которой нужно построить, символ – неизвестное (он, понятное дело, должен быть единственным неопределенным символом, входящим в выражение), а начало и конец задают отрезок оси X для построения графика; участок по оси Y в таком варианте записи выбирается автоматически, исходя из минимума и максимума функции на заданном промежутке. Обратите внимание, что неизвестное и концы промежутка нужно задавать не тремя отдельными параметрами, как, скажем, в integrate, а в виде списка. Это связано с тем, что plot2d может принимать еще и дополнительные аргументы – в таком случае они перечисляются следом за таким списком, что исключает всякую путаницу.

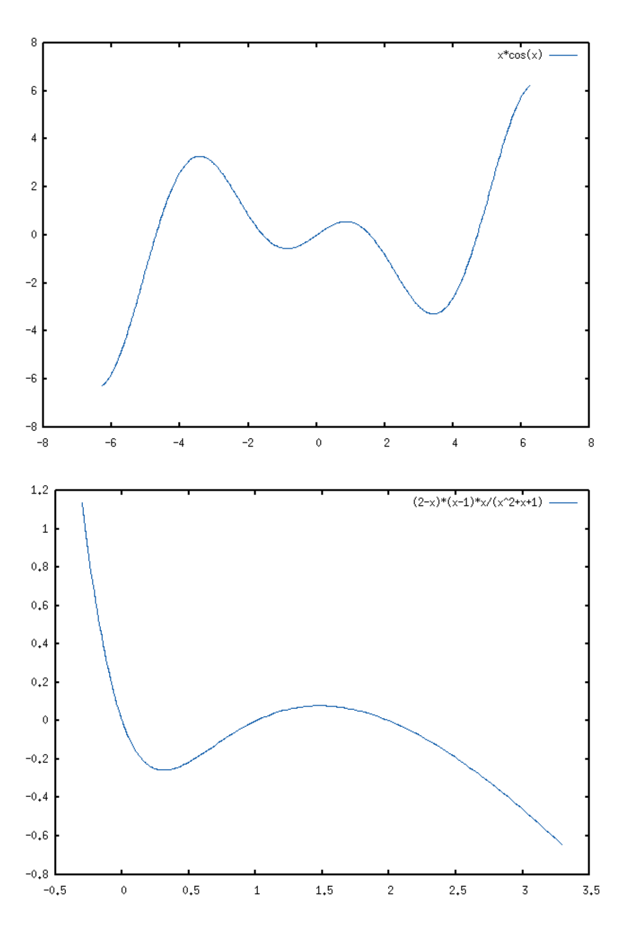
После вызова функции plot2d в таком варианте откроется окно *gnuplot*, в котором будет отображен затребованный график. Никакой интерактивной работы с полученным изображением *gnuplot* не предусматривает, кроме автоматического его масштабирования при изменении размеров окна. Насмотревшись вдоволь, можно закрыть окно с графиком клавишей Q, либо, в случае работы с *Maxima* в редакторе *TeXmacs* или wx*Maxima*, просто переключиться обратно в интерфейс, оставив окно *gnuplot* открытым, и продолжить работу:

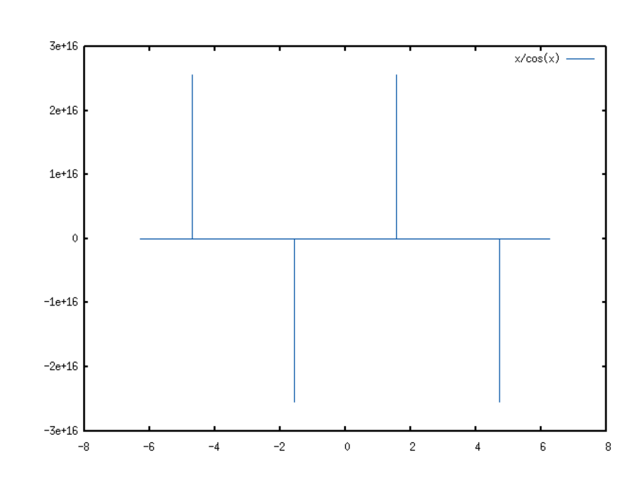
(%i1) $\mathrm{plot2d}(x\cos(x), [x, -2\pi, 2\pi])\$$

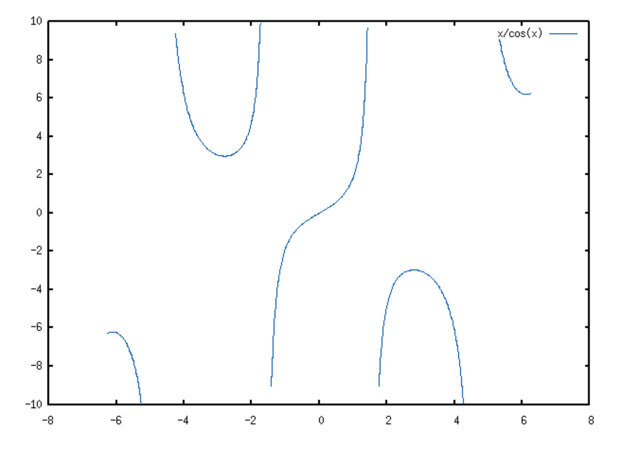
(%i2) $\mathrm{plot2d}\left(\dfrac{x\,(x-1)\,(2-x)}{x^2+x+1}, [x, -0.3, 3.3]\right)\$$

Месяц назад Мы прошли экспресс-курс математического анализа и дифуравнений.

конструкции







В некоторых случаях автоматический подбор отображаемого участка вертикальной оси может нас не устроить. Например, он работает не очень хорошо, если функция имеет на заданном промежутке точку разрыва, хотя бы один из односторонних пределов в которой равен бесконечности: тогда промежуток по оси Y будет выбран слишком большим. Да и в других случаях может понадобиться изменить умолчательное поведение. Для этого предусмотрен такой вариант вызова функции: plot2d(выражение, [символ, начало, конец], [y, начало, конец]). Здесь буква y используется в качестве обозначения вертикальной оси, а остальные два параметра имеют тот же смысл, что и выше.

($%i1) $\text{plot2}d\left(\frac{x}{\cos(x)}, [x, -2\pi, 2\pi]\right)\$$

($%i2) $\text{plot2}d\left(\frac{x}{\cos(x)}, [x, -2\pi, 2\pi], [y, -10, 10]\right)\$$

Как видите, умолчательный вид графиков в *gnuplot* достаточно прост и даже аскетичен, но здесь можно очень и очень многое менять с помощью дополнительных опций. Некоторые из которых будут освещены чуть ниже, а остальные можно изучить по документации к *gnuplot*.

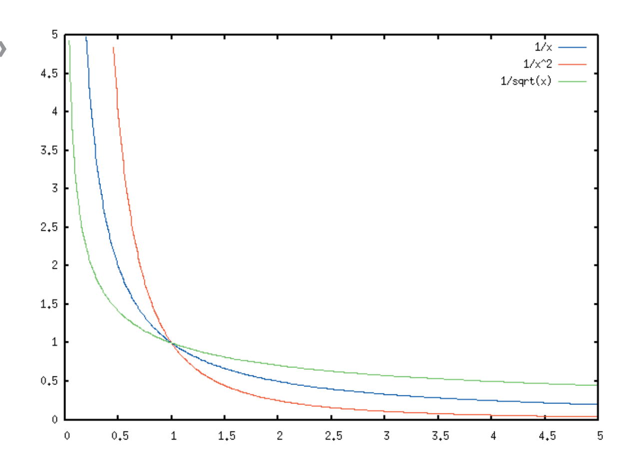
Чтобы построить на одной и той же картинке одновременно два графика (или больше), просто передайте функции plot2d вместо отдельного выражения их список:

($%i1) $\text{plot2}d\left(\left[\frac{1}{x}, \frac{1}{x^2}, \frac{1}{\sqrt{x}}\right], [x, 0.01, 5], [y, 0, 5]\right)\$$

»

Скорая помощь

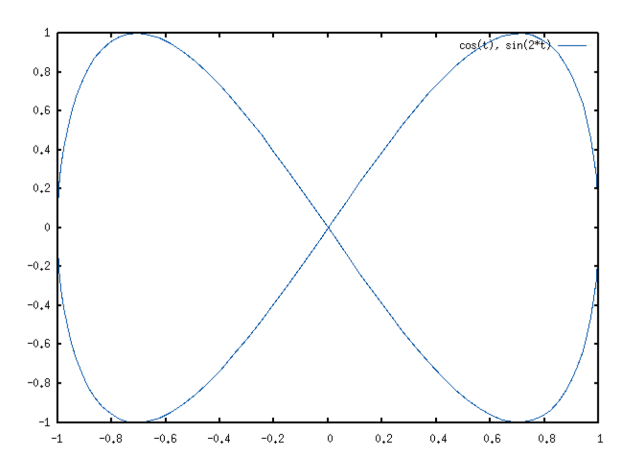
При построении дискретного множества мы использовали две новых функции. Функция random (натуральное число) возвращает псевдослучайное целое число в диапазоне от нуля до заданного. А функция makelist (выражение, переменная, начало, конец) возвращает список из значений «выражения» по всем целым значениям «переменной» от «начала» до «конца».



Здесь [x, 0.01, 5] вместо [x, 0, 5] я написал «по привычке» – *Maxima 5.9.x* выдавала ошибку, если заданная функция была не определена на одном из концов интервала. В 5.10.0 мне эту ошибку воспроизвести не удалось; так что есть основания полагать, что поведение в таких случаях поправили.

Может plot2d строить и графики параметрически заданных функций. Для этого используется список с ключевым словом parametric: plot2d([parametric, x-выражение, y-выражение, [переменная, начало, конец], [nticks, количество]]). Здесь «x-выражение» и «y-выражение» задают зависимость координат от параметра, то есть, по сути, это две функции вида x(t), y(t), где t – переменная параметризации. Эта же переменная должна фигурировать в следующем аргументе-списке, а параметры «начало», «конец», как и в двух других рассмотренных случаях, задают отрезок, в пределах которого этот параметр будет изменяться. Последний аргумент-список, с ключевым словом nticks, задает количество кусочков, на которые будет разбит интервал изменения параметра при построении графика. Этот аргумент опционален, но на практике он нужен почти всегда: умолчательное значение nticks равно 10; согласитесь, редко бывает нужно в качестве графика получить ломаную из 10 отрезков. Вот пример построения графика параметрической функции:

(%i1) $\mathrm{plot2}d([\mathrm{parametric}, \cos(t), \sin(2\,t), [t, -\pi, \pi], [\mathrm{nticks}, 120]])\$$



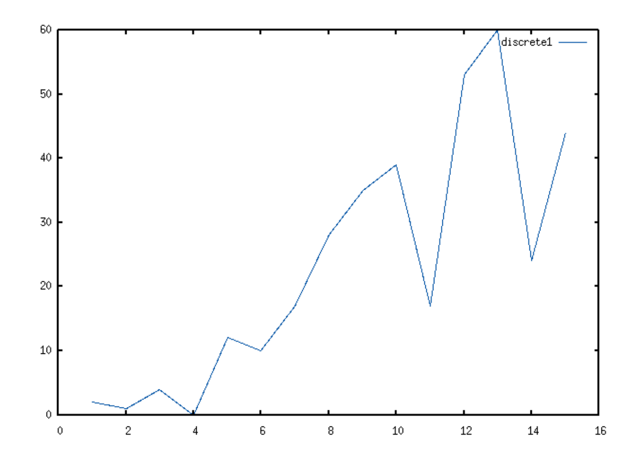
Кроме parametric, функция plot2d понимает еще одно ключевое слово: discrete. Предназначено оно, как нетрудно догадаться, для отображения на плоскости дискретных множеств; точнее говоря, конечных наборов точек. По записи аргументов такой вариант распадается еще на два: plot2d([discrete, x-список, y-список]) и plot2d([discrete, [x, y]-список]). В первом варианте координаты задаются как два отдельных списка [x1, x2, ..., xn], [y1, y2, ..., yn], а во втором – как список пар координат отдельных точек [[x1, y1], [x2, y2], ..., [xn, yn]].

Если мы, к примеру, имеем набор статистических значений, зависящих от номера, мы можем отобразить его, задав в качестве x-координат сами эти номера, то есть натуральные числа:

(%i1) $\mathrm{data}: \mathrm{makelist}(\mathrm{random}(5\,x), x, 1, 15)$

(%o1) $[2, 1, 4, 0, 12, 10, 17, 28, 35, 39, 17, 53, 60, 24, 44]$

(%i2) $\mathrm{plot2}d([\mathrm{discrete}, \mathrm{makelist}(x, x, 1, 15), \mathrm{data}])\$$

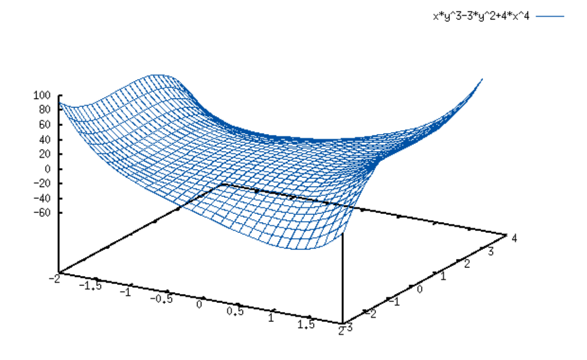


По умолчанию множество отображается в виде ломаной с вершинами в заданных точках; такое поведение можно изменить и получить вывод, к примеру, в виде отдельных точек. Это достигается использованием специальных опций, применимых как к plot2d, так и к plot3d, поэтому давайте перейдем к рассмотрению последней.

Придаем объем

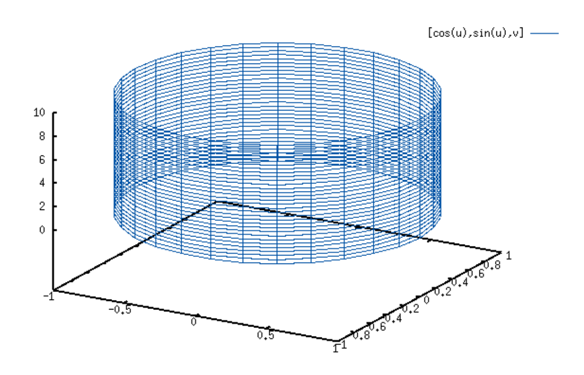
Функция plot3d имеет два варианта вызова: один для явного задания функции и один для параметрического. В обоих случаях функция принимает три аргумента. Для явно заданной функции: plot3d(выражение, [переменная1, начало, конец], [переменная2, начало, конец]); аргументы аналогичны plot2d, с той разницей, что здесь независимых переменных две.

(%i1) $\mathrm{plot3}d(4\,x^4 + x\,y^3 - 3\,y^2, [x, -2, 2], [y, -3, 3])\$$



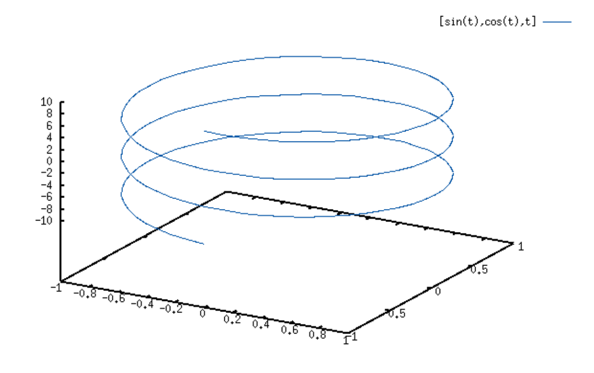
Построение нескольких поверхностей на одном графике не поддерживается – потому, вероятно, что на таком рисунке проблематично было бы что-либо разглядеть. Посему для параметрически заданной функции ключевое слово parametric не требуется: вызов с первым аргументом-списком уже не с чем перепутать. График параметрически заданной функции строится так: plot3d([выражение1, выражение2, выражение3], [переменная1, начало, конец], [переменная2, начало, конец]), где «выражения» отвечают, по порядку, x(u, v), y(u, v), z(u, v).

(%i1) $\mathrm{plot3}d\,([\cos(u), \sin(u), v], [u, -\pi, \pi], [v, 0, 10])\$$



С помощью параметрической формы можно строить и пространственные кривые. Для этого просто нужно задать второй, фиктивный, параметр, чтобы *Maxima* не ругалась на неправильный синтаксис вызова функции:

(%i1) $\mathrm{plot3}d([\sin(t),\cos(t),t],[t,-3\pi,3\pi],[v,0,1],[grid,150,150])\$$



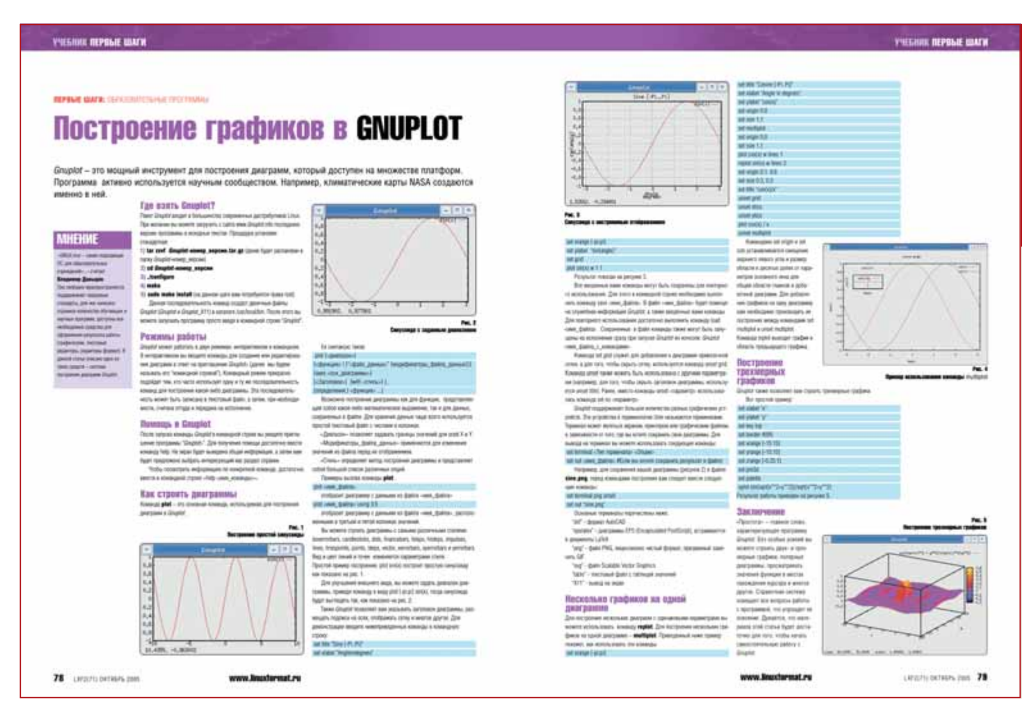
И отсюда мы плавно переходим к опциям функций построения графиков, посредством использованной выше опции grid. Каждая опция имеет некоторое умолчательное значение, а изменить его можно, добавив к аргументам список вида [имя-опции, значение]. Строго говоря, рассмотренные выше y и nticks также являются опциями; в предпоследнем примере мы задали опции nticks значение *120*, а в примере перед ним в качестве значения опции y использовалась пара чисел *0, 5*. В документации к *Maxima* символ x, выступавший в примерах выше в качестве обязательного параметра, также приводится как опция; на самом деле опцией он является только в случае parametric и действует тогда так же, как и опция y, только по другой оси. Опция grid, использованная выше, применима к трехмерным графикам вместо опции nticks, используемой для двумерных. Она, также как и y, задается в виде двух целых значений, которые для поверхностей задают размер ячеек сетки, в виде которой отображается поверхность; первое число – вдоль оси X, второе – вдоль оси Y; либо, в случае параметрического задания, по первому и по второму параметру соответственно. Для кривых из этих параметров действует только один, но писать нужно опять же оба, дабы не нарушать синтаксис; и здесь этот параметр имеет в точности тот же смысл, что nticks для кривых на плоскости. Но перейдем к другим опциям.

С претензией на красоту

Первая опция, которую мы рассмотрим, задает формат вывода результата; так она и называется: plot_format. Формат может принимать одно из четырех значений, первое из которых действует по умолчанию: gnuplot, mgnuplot, openmath и ps. В умолчательном варианте (значение gnuplot) данные для отображения передаются напрямую программе *gnuplot*, которая сама по себе имеет достаточно гибкое управление, и параметры ей можно передавать прямо из *Maxima* с помощью дополнительных опций функций plot2d/3d. Параметров этих настолько много, что *gnuplot* могла бы стать темой отдельной статьи; так что обращайтесь за ними к документации по *gnuplot*. В противовес своим богатым возможностям, *gnuplot* имеет перед следующими двумя интерфейсами (если откровенно – скорее, лишь перед одним из них) только один недостаток: она генерирует статичное изображение, тогда как *mgnuplot* и *openmath* позволяют в реальном времени масштабировать и передвигать картинку, а plot3d – еще и вращать линию или поверхность в разные стороны в пространстве.

Следующий вариант – *mgnuplot* – является дополнительным интерфейсом к *gnuplot*, написанным на Tcl/Tk, но динамика у него настолько «задумчивая», а остальные возможности настолько бедны, что я не вижу смысла останавливаться на нем подробнее.

И перехожу сразу к *openmath*. Он тоже не очень-то поддается управлению, зато предоставляет хорошую интерактивность, особенно ценную в трехмерном варианте: после того, как объект сгенерирован, его можно масштабировать и очень динамично вращать, разглядывая


УЧЕБНИК ПЕРВЫЕ ШАГИ

Построение графиков в GNUPLOT

Мы об этом уже рассказывали: *Gnuplot*

Статья о *Gnuplot* в LXF71 была первым оригинальным материалом российской редакции LinuxFormat.

Владимир Давыдов продемонстрировал основные возможности пакета и привел несколько наглядных примеров, доказывающих, что «простота – главное слово, характеризующее программу *Gnuplot*».

LXF71 можно приобрести в интернет-магазине Линуксцентра отдельно:

http://www.linuxcenter.ru/?good=1279

или в составе пакета из всех 4 номеров, вышедших в 2005 году:

http://www.linuxcenter.ru/?good=1328

И снова о решении проблемы с запуском из TeXmacs

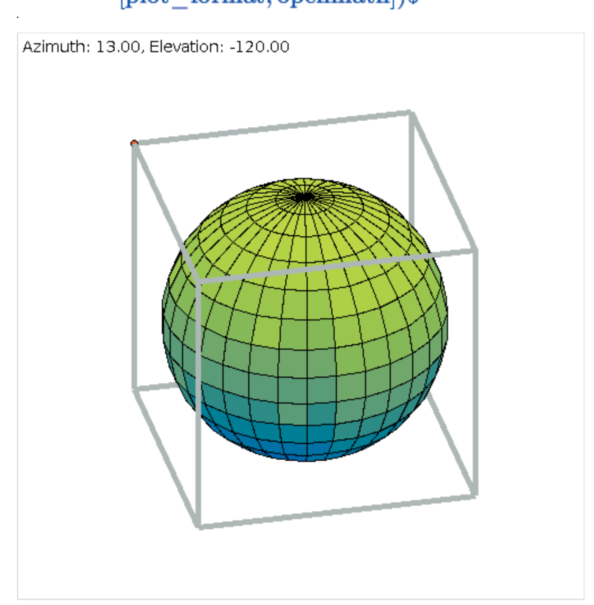
В первой после выхода *Maxima 5.10.0* версии редактора *TeXmacs* есть проблема с запуском конкретно этой версии *Maxima*. Связана она с тем, что скрипт *TeXmacs*, отвечающий за инициализацию Maxima-сессии, проверяет версию *Maxima* на равенство «5.10», в то время как скрипт, выясняющий версию *Maxima*, – **/usr/lib/texmacs/TeXmacs/bin/maxima_detect** – возвращает ее все же как «5.10.0»; в результате чего *Maxima* не запускается, а просто выдается сообщение «Unsupported version of maxima». Решить эту проблему не просто, а очень просто: нужно отредактировать файл **/usr/lib/texmacs/TeXmacs/bin/tm_maxima**, а именно строку, содержащую 5.9.1.1* | 5.9.2* | 5.9.3* | 5.10, дописав после 5.10 точно такую же звездочку. И только после этого запускать *Maxima*-сессию в *TeXmacs* (сам редактор для этого перезапускать не нужно).

со всех сторон. Особенно это помогает для сложных поверхностей, когда, глядя на статичную «сетку» *gnuplot*, непросто понять форму поверхности. Справедливости ради нужно отметить, что *gnuplot* позволяет задавать точку обзора трехмерного объекта в качестве одного из многочисленных параметров, то есть хотя картинка и статична, но с какой стороны на нее смотреть, мы можем указать произвольно.

Ну и последнее значение опции plot_format подталкивает *Maxima* к непосредственной генерации PostScript-документа с изображением. Но и здесь надо сказать: генерировать PostScript-вывод умеет и все тот же *gnuplot*.

Большинство остальных опций относятся только к формату вывода *gnuplot*. А мы рассмотрим еще одну универсальную, пригодную для всех форматов и преобразующую не результирующее изображение, а сам процесс построения графика; точнее, систему координат. Называется эта опция transform_xy, по умолчанию она равна *false*. Передавать ей нужно выражение, сгенерированное функцией make_transform([x, y, z], f1(x, y, z), f2(x, y, z), f3(x, y, z)). Кроме того, существует одно встроенное преобразование, известное как polar_xy и соответствующее make_transform([r, th, z], r*cos(th), r*sin(th), z), то есть переходу к полярной цилиндрической системе координат. В качестве примера использования transform_xy приведу преобразование к полярным сферическим координатам, раз уж во встроенном виде его нет:

(%i1) $\mathrm{plot3}d(1,[f,0,2\pi],[p,0,2\pi],[\mathrm{transform_xy},\mathrm{make_transform}([t,f,r], r\sin(f)\sin(t), r\cos(f)\sin(t), r\cos(t))],[\mathrm{plot_format},\mathrm{openmath}])\$$



Обратите внимание: в первом аргументе-списке к make_transform последним должен идти зависимый символ, то есть тот, который будет выступать функцией от двух других.

Если вам нужно постоянно работать со сферическими координатами, можете задать, скажем, spherical_xy:make_transform([t, f, r], r*sin(f)*sin(t), r*cos(f)*sin(t), r*cos(t)), и затем при построении графиков писать [transform_xy, spherical_xy].

Ветвитесь и повторяйтесь

До сих пор мы двигались только по прямой, а теперь поговорим о средствах «изменения траектории»: условном операторе и циклах.

Начнем с условия. В *Maxima*, в отличие от большинства «традиционных» процедурных и объектных языков программирования, где существует так называемый условный оператор, привычная связка if-then-else является не синтаксической конструкцией, а самым настоящим оператором. По своему действию он больше всего похож на тернарный оператор языка C, только с более «человеческим» синтаксисом: if условие then выражение1 else выражение2. При выполнении «условия» из двух «выражений» вычисляется только первое и возвращается как результат оператора; в противном случае выполняется только второе и оно же является значением всего выражения if-then-else. Часть конструкции else выражение2, как и в большинстве языков программирования, опциональна. Если ее нет, а условие все-таки не выполнилось, результат оператора if будет равен *false*.

При этом, конечно же, никто вам не мешает использовать этот оператор как обычную условную конструкцию, а возвращаемое значение просто игнорировать. С другой стороны, оператор if можно применять, например, для задания рекурсивных последовательностей:

(%i1) $a(n):=\mathrm{if}\ n\leqslant 1\ \mathrm{then}\ n\ \mathrm{else}\ \frac{a(n-1)+a(n-2)}{2}\$$

(%i2) $a(20)$

(%o2) $\frac{349525}{524288}$

(%i3) $\%,\mathrm{numer}$

(%o3) 0.66666603088379

Немного о самих условиях, которые могут проверяться оператором if. Условия >, <, >=, <= записываются и расшифровываются традиционно, так же как и логические операторы and, or, not. А вот о равенствах-неравенствах нужно сказать пару слов. Равенство в *Maxima* есть двух видов: синтаксическое и логическое. Знаком = обозначается как раз первое, а второе вычисляется с помощью функции equal(). Чтобы не быть многословными, отличие синтаксического равенства от логического продемонстрируем на примере; здесь дополнительно используется предикат по имени is, которые проверяет на истинность свой аргумент.

(%i1) $\mathrm{is}\left((x+1)^2=x^2+2x+1\right)$

(%o1) **false**

(%i2) $\mathrm{is}\left(\mathrm{equal}\left((x+1)^2,x^2+2x+1\right)\right)$

(%o2) **true**

Ну и неравенств, соответственно, тоже существует два, с тем же смыслом. Синтаксическое неравенство обозначается достаточно непривычно – через #; видимо, этот символ разработчики сочли наиболее визуально схожим со знаком ≠. Ну а логическое неравенство обозначено через notequal().

Конечно, кроме упомянутых сравнений в условном операторе можно использовать любые предикаты, то есть функции, возвращающие логические значения *true/false*. Функций таких достаточно много, но все они достаточно просты, поэтому не буду тратить время на их описание: его можно почерпнуть в том же объеме из документации.

Напоследок перейдем к циклам. Цикл в *Maxima* будто бы тоже один. Но он имеет столько различных вариантов, что назвать это все одним оператором цикла язык не поворачивается. Вот как выглядят основные разновидности:

for переменная:начало step шаг thru конец do выражение
for переменная:начало step шаг while условие do выражение
for переменная:начало step шаг unless условие do выражение

Первый прокручивает цикл, изменяя «переменную» с заданным «шагом» от «начала» до «конца»; второй – от «начала» и пока выполняется «условие»; третий – наоборот, пока «условие» не выполняется. К примеру, мы можем получить список из первых десяти членов последовательности из позапрошлого примера:

(%i1) $a(n):=\text{if } n\leqslant 1 \text{ then } n \text{ else } \dfrac{a(n-1)+a(n-2)}{2}\$$

(%i2) $A:[]\$\text{for } n:0 \text{ thru } 10 \text{ do } A:\text{append}(A,[a(n)])$

(%o3) done

(%i4) A

(%o4) $\left[0,1,\frac{1}{2},\frac{3}{4},\frac{5}{8},\frac{11}{16},\frac{21}{32},\frac{43}{64},\frac{85}{128},\frac{171}{256},\frac{341}{512}\right]$

(%i5) A,numer

(%o5) $[0,1,0.5,0.75,0.625,0.6875,0.65625,0.671875,0.6640625,0.66796875,0.666015625]$

Как видите, в качестве оператора цикл в простейшем его виде, в отличие от условия, использовать смысла нет, так как его возвращаемое значение всегда равно done. В этом примере один из элементов циклического оператора не указан; шаг, как видите, может быть опущен и по умолчанию равен единице. Самое интересное в этом операторе то, что опустить позволяется любую его часть, кроме do; и в том числе в любых комбинациях. К примеру, опустив кроме step еще и for, мы получаем из этого же оператора традиционные циклы while и unless (второй и третий варианты). А проделав то же самое с первым вариантом записи, получим цикл без счетчика вида thru число do выражение, который просто повторится заданное число раз. Можно, наоборот, опустить условие окончания и получить цикл с индексной переменной, но бесконечный. А оставив только do, получим самый простой вариант бесконечного цикла. Из таких бесконечных циклов можно выйти с помощью оператора return(выражение) (точнее, конечно, конструкции из двух операторов вида if условие then return(выражение)), который прервет выполнение цикла и вместо done вернет заданное выражение. Естественно, оператор return() можно применять во всех видах циклов, а не только в бесконечных.

Но и это еще не все. Кроме всех уже рассмотренных вариаций, цикл может принимать еще две ипостаси. Во-первых, вместо step может использоваться конструкция next выражение, смысл которой лучше тоже продемонстрировать на примере

(%i1) $\text{for } i:162304 \text{ next } \frac{i}{2} \text{ while integerp}(i) \text{ do display}(i)\$$

i = 162304
i = 81152
i = 40576
i = 20288
i = 10144
i = 5072
i = 2536
i = 1268
i = 634
i = 317

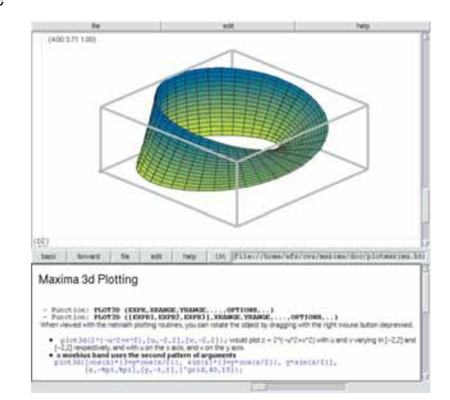
После next может стоять любое вычислимое выражение относительно индекса цикла, и применяться эта конструкция может во всех трех вариантах цикла (thru/while/unless).

А «во-вторых» – это еще один отдельный вариант цикла: for переменная in список do выражение; либо расширенная форма: for переменная in список условие do выражение. Здесь цикл будет прокручен с «переменной», изменяющейся по всем элементам «списка»; плюс можно задать еще и дополнительное «условие» на прерывание цикла. Вот теперь мы с циклами действительно закончили. Как видите, все достаточно разнообразно. Я, признаться, ничего, что здесь не реализовано, и придумать не смог.

Но рассказ о циклах и условном операторе остается неполным, пока я не рассказал о группировке выражений – ведь в обычном варианте после then или do можно написать всего одно из них. А группировка, или, как ее принято называть, составной оператор в *Maxima* – это опять-таки самый настоящий оператор, который тоже, как и положено оператору, возвращает некоторое значение. Обозначается он скобками, самыми что ни на есть круглыми и обыкновенными; а разделяются сгруппированные операторы/выражения внутри этих скобок не менее обыкновенными запятыми. Возвращаемым значением составного оператора является последнее вычисленное выражение.

С условным оператором, столь разнообразными циклами и составным оператором мы уже можем, комбинируя их между собой и с любыми другими функциями и выражениями *Maxima*, писать полноценные программы с использованием богатого символьного математического аппарата. Естественно, теперь нам захочется сохранять эти программы в виде внешних файлов, чтобы не набирать их каждый раз вручную, а подгружать одной короткой командой. Об этом, а также о математических аналогах объявления переменных – в завершающей статье цикла.

Мы также поговорим о математических аналогах объявления переменных и рассмотрим практические примеры с применением уже достаточно богатого известного нам инструментария. LXF



Скорость обновления Maxima

Существование свободной программы в такой сложной и профессиональной области, как символьная математика, конечно, само по себе не может не радовать. Но вместе с тем, на данный момент *Maxima* во многих направлениях заметно отстает от коммерческих приложений этого же назначения – *Mathematica* и *Maple*, в развитие которых уже много лет подряд вкладываются большие деньги. На этом фоне несколько лет кряду выглядели неутешительно и темпы развития самой *Maxima*. По сравнению с упомянутыми коммерческими приложениями сроки выпуска новых версий, конечно, более чем на уровне. Но как для открытого проекта – это не те темпы, которые были бы поводом для счастья. Судите сами: дата выхода версии 5.9.0 – 10.02.2003, 5.9.1 – 24.09.2004, 5.9.2 – 17.10.2005. Итого, больше года между соседними релизами.

И вот за последний год время между выпусками сократилось минимум вдвое, и таким образом скорость выпуска вышла на хороший уровень активно развивающегося FOSS-проекта. Начиная этот цикл, я еще считал «почти новой» (по старым меркам) версию 5.9.2: ей тогда было немногим больше полугода; хотя 5.9.3 за эти пол-года уже успела выйти – 21 марта. И вот сейчас (по состоянию на начало октября) уже доступна следующая «мажорная» версия – 5.10.0, увидевшая свет 23 сентября. И это дает основания надеяться, что с нынешнего момента *Maxima* все же начнет догонять дорогие закрытые продукты по основным возможностям.

Через месяц
Работа с файлами и практические примеры использования *Maxima*.




Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru

1 Вопросы к SUSE

В Недавно установил SUSE, и у меня есть несколько вопросов. Во-первых, куда устанавливать дополнительные модули для *Firefox*? Директории **mozilla/plugins** найти не могу. Во-вторых, при установке я создал пользователя, но у него нет root-привилегий. Как это поправить?

Брайан Клифтон [Brian Clifton]

О Дополнительные модули для *Mozilla* и *Firefox* помещаются в одно из двух мест, в зависимости от типа установки: от имени обычного или суперпользователя. Общесистемные модули и расширения попадают в **/usr/lib/Firefox/plugins** и **/usr/lib/Firefox/extensions** соответственно. Те модули и расширения, что Вы устанавливаете под пользователем прямо с сайтов вроде http://plugindoc.mozdev.org или http://addons.Mozilla.org, сохраняются внутри Вашей домашней директории. А именно: в каталоге **.Mozilla/Firefox/xxx.default**, где **xxx** – случайно сгенерированная строка символов. Вообще-то эти файлы лучше не трогать; устанавливать, удалять и обновлять расширения можно прямо из *Firefox*.

Создаваемый при установке пользователь не имеет и не должен иметь привилегий root: иначе какой смысл в существовании суперпользователя? Когда какой-либо программе, например, *YaST*, потребуются привилегии root, она спросит у Вас пароль суперпользователя, заданный Вами при установке. Отработав с привилегиями root столько, сколько нужно, программа вернется в обычный режим. Если Вы хотите запустить команду в терминале от лица суперпользователя, наберите:

```
su -c "команда, которую Вы хотите запустить"
```

для одной команды или же

```
su -
первая_команда
вторая_команда
...
logout
```

для нескольких. В обоих случаях потребуется ввести пароль суперпользователя. **НБ**



› **Можно установить модули для *Firefox* вручную, но лучше делать это с соответствующих web-сайтов.**

2 Установка CentOS

В Я установил CentOS 4.3 на свой Pentium III с Windows 98, и все работает превосходно. В LXF81 написано, что для CentOS нужен процессор Pentium, а у меня еще есть AMD FX-53 с Windows XP и Fedora 5 – хотел бы заменить на нем Fedora на CentOS, но боюсь пробовать. Можно ли установить CentOS на компьютер AMD? Если нет, нельзя ли это сделать с помощью какого-нибудь стороннего ПО? И планируется ли на будущее версия для AMD?

Р. Дэвисон [R.Davison]

О На самом деле было написано «процессор класса Pentium», то есть все совместимое с i586. Pentium – торговая марка Intel, но процессоры AMD вполне совместимы. Запустить данную версию CentOS на Вашем FX-53 можно, но процессор будет недогружен: он 64-разрядный, а для запуска CentOS 4.3 переключится в 32-битный режим. Работать он будет все равно быстрее, чем обычные 32-битные процессоры, однако для получения оптимальной производительности лучше скачать с сайта www.centos.org 64-разрядную версию дистрибутива. Если у Вас нет возможности его скачать, можете заказать дистрибутив в Интернет-магазине или переписать у кого-нибудь из местной группы пользователей Linux. **НБ**

Наши эксперты

» Мы найдем эксперта на любой вопрос! Вы получите ответ на все: от проблем с установкой или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Кингс Кобблер
Кингс – системный инженер Linux и администратор Rackspace, использует Linux десять лет, всегда готов отвечать на письма других администраторов.



Ник Вейч
В свободное от исчеркивания текстов красными чернилами время Ник возится с Linux-графикой и 3D-приложениями; он у нас отвечает за простые вопросы!

КУДА ПОСЫЛАТЬ ВОПРОСЫ:
Пишите на м по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

3 Зависание

В Пытаюсь смонтировать NFS-директорию, находящуюся на сервере, на своей рабочей машину, и все зависает. В журналах никаких указаний нет. В чем тут причина?

Хорли [Horley]

О NFS полагается на RPC (Remote Procedure Call, удаленный вызовов процедур). Ключ ко всему этому в сервисе ***portmap***, который обрабатывает RPC-запросы и устанавливает соединение с нужной процедурой. Проверьте, что portmap запущен:

```
[root@test gnump3d-2.9.8]# ps -ef | grep portmap
rpc   2584 1 0 Jul23 ?       00:00:00 portmap
root 30843 30474 0 07:45 pts/4 00:00:00 grep portmap
```

Далее Вам нужно убедиться, что работают следующие RPC: *mountd, nfsd, lockd, statd, lockd, rquotad* и *imapd*. В зависимости от дистрибутива, они могут быть запущены из соответствующего стартового скрип-

» та NFS. Если они не работают, их можно запустить вручную:

```
[root@test]# rpc.mountd
[root@test]# ps -ef | grep mountd
root 30906 1 0 07:54 ?      00:00:00 rpc.mountd
```

Вы можете также проделать это с остальными сервисами, но понадобится вписать их запуск в стартовый скрипт, чтобы они обязательно перезапускались при загрузке системы. **КК**

4 Молчание – не золото

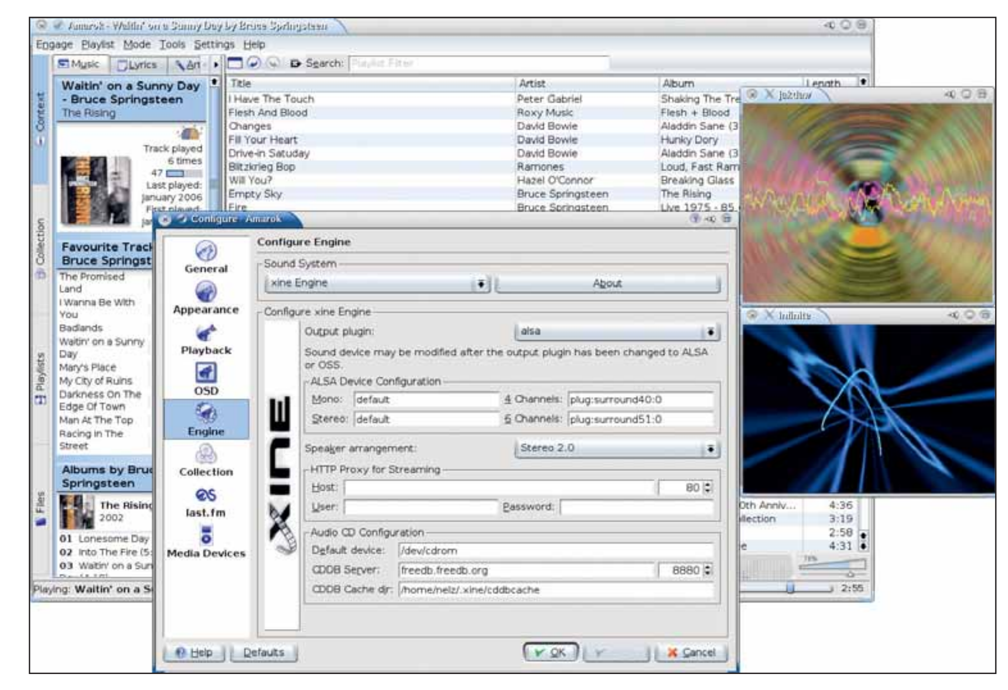
Я – новичок в Linux и только недавно установил SUSE 10.1. Когда я пытаюсь запустить *Amarok*, в нем нет звука, хотя звуковая карта работает (при загрузке слышен звуковой эффект).

Я увидел, что на вашем диске (к LXF82) есть новая версия (1.4), и попробовал установить ее через *YaST*. Вы не могли бы рассказать, как установить новую программу с DVD к LXF через *YaST*? Понимаю, что мой вопрос, наверное, очень простой, но я просто не знаю, что делать.

Эдриен [Adrian]

Здесь два разных вопроса: по *Amarok* и по установке программ.

Посмотрите на строку состояния *Amarok*, когда пытаетесь проиграть файл – это даст вам наводку. Если песня идет, но звука не слышно, откройте микшер (обычно иконка колонки в панели задач) и убедитесь, что выставлен нужный уровень громкости. Если *Amarok* не проигрывает песню, проблема, возможно, в настройке движка. Посмотрите на секцию Engine в окне настроек программы: если там значится 'aRts', а рабочий стол у Вас Gnome, Вы вряд ли что-нибудь услышите, поскольку *Arts* – звуковой движок среды KDE. Лучшим выбором, и по качеству звука и по совместимости с обеими средами, является *Xine*. Возможно, потребуется также указать модуль вывода; Autodetect обычно работает, но если нет – укажите ALSA.



› *Amarok* – отличный аудиоплейер, особенно если музыка слышна! Проверьте, что в настройках указаны подходящие движок, звуковое устройство и уровень громкости.

YaST больше подходит для установки программ из известных ему репозитариев: например, из директории SUSE на установочном диске и онлайн-каталогов обновления пакетов, указанных при установке автоматически или впоследствии вручную. Вы можете велеть *YaST* устанавливать отдельные RPM-пакеты; для установки пакетов с DVD выполните в терминале от имени суперпользователя:

```
su
YaST2 --install /media/LXFDVD82/Sound/Amarok/
    SUSE/*.rpm
```

Правда, при таком запуске *YaST* не обрабатывает зависимости, и может рухнуть, причем без указания причин. Лучше использовать команду rpm:

```
su
rpm -Uhv /media/LXFDVD82/Sound/Amarok/
    SUSE/*.rpm
```

Установка пакетов может и не удасться, зато Вы будете знать, в чем дело. Более правильное решение – добавить в *YaST* репозитории с дополнительным ПО. Сайт со списком репозиториев и инструкциями по их добавлению – http://en.opensuse.org/Additional_YaST_Package_Repositories. **НБ**

Краткая справка по ...

sudo

Root с ограничениями. Советы по использованию этой утилиты.

Безопасность работы в Linux отчасти обусловлена тем, что пользователям не дано административных привилегий. Иногда, правда, привилегии бывают нужны – например, чтобы установить программу или поправить конфигурационный файл. Тут уж приходится переключаться на суперпользователя (root). Традиционно это делает команда su (switch user): она предоставляет пользователю администраторский доступ, и пока он работает в системе с правами root, то способен на полный беспредел. Su также требует знания пароля суперпользователя. А не безопаснее ли разрешить пользователям запускать от имени root только определенные команды?

Именно это и делает *sudo*. Файл конфигурации **/etc/sudoers** содержит перечень команд, которые пользователь может выполнять как root. Запуск

```
sudo <определенная_команда>
```

приведет к запросу пароля (пароля того, кто запустил команду, а не суперпользователя, так что нет нужды разглашать последний), и команда выполнится, только если она дозволена пользователю; таким образом, безопасность системы резко возрастает.

Типичная запись в **/etc/sudoers** выглядит примерно так:

```
nelz ALL = /bin/mount,/bin/ umount
```

Здесь пользователю nelz дается право запускать mount и umount на всех хостах. Запрос пароля можно убрать, следующим образом:

```
nelz ALL = NOPASSWD: /bin/mount,/bin/umount
```

Будьте с этим осторожны! Можно определять псевдонимы для групп пользователей, команд или хостов, чтобы упростить администрирование. Не редактируйте **/etc/sudoers** напрямую; используйте команду visudo.

Несмотря на свое имя, она запускает не обязательно Vi, а тот редактор, который прописан в переменной окружения **$EDITOR**. *Visudo* проверяет синтаксис настроек перед записью в **/etc/sudoers**, чтобы вы не заблокировали себя из-за случайной ошибки.

5 Логика логгинга

Можете ли вы мне помочь уменьшить количество одинаковых сообщений в журналах? Когда я только начал использовать Linux, в log-файлах бывали записи вроде «сообщение повторяется x раз», но это стало редкостью. Проблема не столько в размерах файлов журналов, сколько в нахождении действительно важных сообщений. Прилагаю ниже несколько частых групп сообщений, получаемых в MEPIS 3.4.

Первая группа приходит от моего ZIP-диска, который пачкает журнал во время загрузки и извлечения. Большая цифра – размер диска, меньшая происходит из таблицы разделов. Вторая группа говорит о чьих-то «пингах» с интервалом в минуту. Тут 10.10.10.134 – локальный IP-адрес, а 10.10.10.91 – удаленный. Третья группа – сотни подобных сообщений, выдающихся за несколько секунд, хотя это случается лишь иногда. Вы можете видеть признаки «гонки» (race condition).

На работу машины это особо не влияет, но зачем мне разгребать весь этот мусор, если произойдет серьезная ошибка?

Вот пример сообщений: »

```
» Jul 18 19:07:40 localhost kernel: hdd: The disk
    reports a capacity of 752896000 bytes, but the
    drive only handles 752877568
Jul 18 19:07:40 localhost kernel: hdd: hdd4
Jul 18 19:13:20 localhost kernel: martian source
    10.10.10.255 from 10.10.10.134, on dev eth1
Jul 18 19:13:20 localhost kernel: ll header: ff:ff:ff:ff:
    ff:ff:00:0a:5e:1d:53:c2:08:00
Jul 18 19:14:00 localhost kernel: [unmap_page_
    range+217/232] unmap_page_range+0xd9/0xe8
Jul 18 19:14:00 localhost kernel: [unmap_
    vmas+172/376] unmap_vmas+0xac/0x178
Jul 18 19:14:00 localhost kernel: [unmap_
    region+125/242] unmap_region+0x7d/0xf2
```

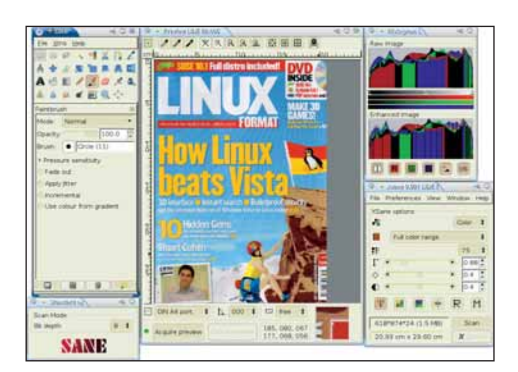
Сесил Уоллис [Cecil Wallis]

Я вижу три подхода к этой проблеме. Первый – найти причину появления этих сообщений и пресечь их появление. Присланный Вами файл **system.txt** очень помог нам узнать точную причину третьей группы сообщений: она в том, что Вы используете ядро 2.6.15 с видеокартой Nvidia. Для решения проблемы нужно обновить ядро или установить SimplyMEPIS 6.0 с нашего диска к LXF84.

«Марсианские» сетевые входы указывают на немаршрутизируемые пакеты, в данном случае, с широковещательным адресом 10.10.10.255. Уведомления можно отменить, выполнив такую команду от лица root:

```
echo "0" >/proc/sys/net/ipv4/ip_log_martians
```

Но сначала следует отыскать причину их появления. Они могут приходить от не сконфигурированного сетевого оборудования или служить признаком атаки на Ваш компьютер. Если они имеют место и при отсутствии доступа в Интернет, значит, проблема в локальной сети; в противном случае проверьте свой брандмауэр.



› **Linux в наши дни поддерживает множество сканеров. Эта модель Canon сканирует прямо в *Gimp*.**

Ошибка Zip-привода может быть неистребима, но тогда применим другой подход: фильтрация журнала, с целью игнорировать незначащие записи. Для избавления от «шума», пропустите лог-файл через *grep*, например, таким образом:

```
grep -v -f /var/log/filter /var/log/messages | less
```

Здесь **/var/log/filter** – файл, содержащий шаблоны лишних записей, по шаблону в строке:

```
localhost kernel: *hdd:
```

Третий подход самый всеобъемлющий, но и самый сложный: настроить систему так, чтобы сообщения отсылались в разные файлы (или вообще в **/dev/null**). В MEPIS используется sysklogd, его функции фильтрования весьма ограниченны. Вместо него можно взять пакет *syslog-ng*, а в его файле конфигурации (**/etc/syslog-ng/syslog-ng.conf**) указать файл, куда писать сообщения, касающиеся жесткого диска или ZIP:

```
destination messages { file("/var/log/messages");};
destination d_zip { file("/var/log/zip"); };
filter f_zip { match("hdd"); };
filter f_nozip { not match("hdd"); };
```

Затем замените строку destination messages { file("/var/log/messages");}; следующим:

```
log { source(src); filter(f_nozip);
    destination(messages); };
log { source(src); filter(f_zip); destination(d_zip); };
```

Первый фильтр распознает все сообщения о жестком диске, они направляются в отдельный файл. Второй обрабатывает все прочие сообщения: эти записываются в журнал. Можно сузить поиск, уточнив шаблон, но он должен быть одинаков для обоих фильтров, иначе Вы будете терять сообщения. **НБ**

6 Печать и сканирование

Я недавно установил SUSE с LXF DVD (OpenSUSE Slick, LXF78). Все работает нормально, но ни из одного приложения ничего не печатается. Мой принтер (HP 1200) распознан корректно, однако при попытке печати задачи просто бесконечно долго висят в очереди. Есть идеи?

Также, я пытаюсь найти планшетный сканер для домашнего использования (не слишком дорогой), способный работать, скажем с Xandros или SUSE. Повидимому, со сканерами в Linux проблема.

Последний вопрос: почему в **/dev** такой беспорядок? Разве нельзя опрашивать устройства и создавать для них файлы автоматически? Дополнительные устройства можно добавить и вручную.

Дэвид Боускилл [David Bowskill]



Часто задаваемые вопросы

Удаленный рабочий стол

Свернуть горы поможет технология удаленного доступа.

» Хочу соединиться с моим домашним компьютером с работы и запустить графические программы. Возможно ли это в Linux?

Да, возможно. Для этого есть два простых способа. Первый – *X-forwarding*, который лучше всего подходит, если на обоих компьютерах установлен Linux. Он работает через SSH-соединение: вы запускаете программу на удаленном компьютере, а она отображается на локальном рабочем столе. Например:

```
ssh -X me@my.home.computer
kmail
```

Опция -X служит для включения X-forwarding.

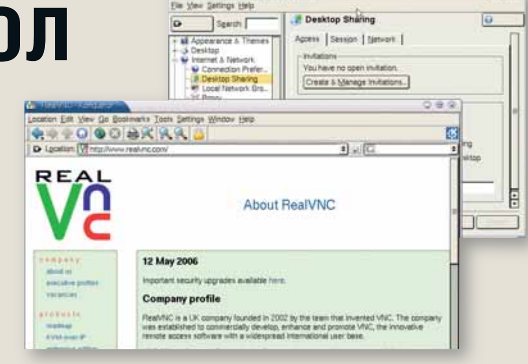
» Это – все, что нужно для запуска программ на удаленной машине?

Строго говоря, программа запускается на удаленной машине, а отображается на локальной. Вам может потребоваться изменить **/etc/ssh/sshd_config**, выставив опцию X11Forwarding в *Yes*.

» Этот метод позволяет отображать только программу. А если я хочу увидеть весь удаленный рабочий стол?

Тогда лучше использовать программу удаленного рабочего стола. Так же следует поступить и если локальная машина работает под Windows. Самое популярное решение – *VNC* (*Virtual Network Computer*). Это клиент-серверное приложение, *VNC*-сервер запускается на удаленной машине, а клиент – на локальной. На www.realvnc.com есть пакеты, собранные для всех популярных операционных систем, и можно получать доступ к Linux-системе из Windows или MacOS и наоборот.

» Уже лучше. Что мне нужно установить?



› **Просмотр удаленного рабочего стола на локальном компьютере возможен благодаря *VNC*.**

Если вы используете KDE, то у вас уже все есть. В KDE есть встроенный RFB, это протокол, используемый *VNC* и совместимыми системами. На другом рабочем столе установите *VNC* или *TightVNC* из репозиториев вашего дистрибутива. Обзор в LXF84 показал, что *TightVNC* – хорошая альтернатива *VNC*.

» Скорость соединения важна?

Отображение графического рабочего стола в его актуальном состоянии требует хорошего соединения, так что модем тут не подойдет. Широкополосное соединение сгодится, но помните, что важна исходящая скорость, а так как она редко превышает 384 Кб/с, то лучшим выбором будет *TightVNC*. Уменьшение разрешения экрана и глубины цвета, а также отключение визуальных эффектов помогут снизить нагрузку на соединение.

» Есть ли альтернативы?

Есть *NX* (www.nomachine.com), проприетарная система, значительно увеличивающая скорость. Это закрытое ПО и запускается не на всем оборудовании, поддерживаемом Linux, но если для вас эти факторы несущественны, рассмотреть его стоит.

О Без дополнительной информации сложно сказать, что не так с Вашим принтером. Работала ли тестовая печать, когда Вы в первый раз его настраивали? Лучший источник информации по ошибкам – файл журнала CUPS. Запустите такую команду в терминале:

```
tail -f /var/log/cups/error_log
```

Если Вы получите сообщение о невозможности прочитать файл, воспользуйтесь *su*, войдите как root и повторите попытку. На терминале появятся сообщения CUPS, они должны помочь Вам найти причину ошибки.

Возможно, принтер просто дезактивирован (такое бывает после ошибки). Тогда просто очистите очередь печати и повторите все снова. Вы можете сделать это через менеджер печати Gnome или KDE, либо из командной строки:

```
/usr/bin/enable PrinterName
```

Это должно быть сделано от лица root, причем обязательно надо указать полный путь к команде.

Поддержка сканеров в Linux на сегодняшний день довольно хороша, благодаря SANE (Scanner Access Now Easy). На сайте проекта (www.sane-project.org) находится полный список поддерживаемых моделей. Если хотите личную рекомендацию, я несколько месяцев назад купил Canon LiDE 60. Он дает хорошее качество картинки и отлично работает с SANE. Поддержки кнопок, находящихся на панели сканера, пока нет, но сканирование из программ дает превосходный результат.

Множество файлов устройств в **/dev** создается по требованию. Подключите сканер, принтер или USB-«брелок», и появятся новые файлы; отключите устройства, и файлы исчезнут. Директория **/dev** выглядит захламленной, поскольку многие файлы используются системой, хотя пользователь может и не подозревать об этом. Статичная директория **/dev** – это норма, но современные Linux-системы используют udev для создания файлов устройств при обнаружении нового оборудования. **НБ**

7 Все на своем месте

В Я наслышан про термин FHS. Что это такое и с чем его едят?

Р.Элиа [R.Elia]

О FHS (Filesystem Hierarchy Standart) – набор рекомендаций по иерархии и содержимому системных директорий в Unix и подобных ОС. Например, сюда включено требование, что «приложения никогда не должны создавать и требовать специфичных файлов или директорий, находящихся в корне (/)»; это для того, чтобы содержимое корневого раздела оставалось простым, наглядным и безопасным.

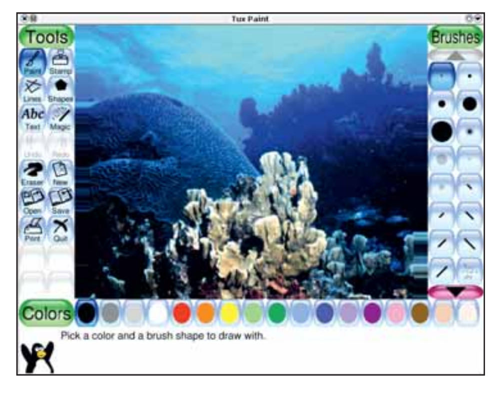
Большинство Linux-дистрибутивов в основном следуют FHS, поэтому у них похожие иерархии файлов. Для каждой директории в FHS определено свое назначение. **/dev** содержит файлы устройств, **/lib** – библиотеки, необходимые для запуска программ из **/bin** и **/sbin**, **/usr** хранит большинство программ и библиотек, с которыми работает пользователь. Так что это ключевая директория в любой Linux-системе.

Короче, FHS необходим, чтобы предотвращать хаос в иерархиях файловых систем различных дистрибутивов Linux. Это значит, что пользователь может увидеть ожидаемые файлы в нужных директориях, а программы могут легко найти нужные им файлы.

Первая иерархия файловой системы для Linux была выпущена в 1994 году. В 1995-м она была расширена, появилась поддержка BSD-систем, и была переименована в FHS. Стандарт поддерживается Free Standards Group, наряду с проектом Linux Standard Base. Хотя все дистрибутивы поддерживают FHS, иерархия в них слегка различается в деталях, например, иногда отсутствуют некоторые директории. Это одна из причин, по которой разные Linux-системы иногда несовместимы между собой. **КК**

8 Архивирование изображений

В Как, используя *mogrify* для изменения размера и формата коллекции изображений, установить целевую директорию для вывода, и еще сделать так, чтобы имя файла содержало временную отметку?



› Фантастический *Tux Paint*. Утилиты *ImageMagick* запросто конвертируют фото в нужный формат.

Я работаю с группой маленьких детей, и это интенсивная работа. Часто я просто открываю *Gimp*, изменяю размер изображения и сохраняю результат в **~/.tuxpaint/saved**, чтобы его можно было использовать в *TuxPaint*. Но тогда детям приходится долго ждать.

Сейчас моя команда выглядит вот так:

```
mogrify -antialias -geometry 448x376 -format png
digicampic.jpg
```

Однако она не помещает изображение в **~/.TuxPaint/saved**, и еще я бы хотел, чтобы команда добавляла в имя файла отметку времени, типа 20060719162549.png.

Лэнсер [Lancer], с форума LXF

О Во-первых, здорово, что Вы обучаете работать в Linux детей «с младых ногтей». Лучше, когда дети растут, понимая, что Windows – не единственный возможный выбор. *mogrify* из комплекта *ImageMagick* модифицирует изображения «на месте», сохранить вывод в другой директории не получится. Для этого подойдет команда convert из все того же *ImageMagick*.

Вот что выполнит нужные Вам действия:

```
for PIC in *.jpg
do
  convert -antialias -resize 448x376 ${PIC}
    $HOME/.TuxPaint/saved/$(date
    +%Y%m%d%H%M%S).png
done
```

Главная проблема тут в том, что следующая картинка может затереть предыдущую, если обе обрабатываются в пределах одной секунды. Можно, конечно, сделать проверку: если имя уже существует, добавить к нему еще одну цифру. Но если вы используете время конвертирования, а не получения изображения, то проще вставить секундную задержку:

```
for PIC in *.jpg
do
  while true
  do
   DEST=$HOME/.TuxPaint/saved/$(date
+%Y%m%d%H%M%S).png
   [ -f ${DEST} ] || break
   sleep 1
  done
  convert -antialias -resize 448x376 ${PIC}
    ${DEST} && mv ${PIC} done/
done
```

Эта версия также перемещает файл в нужную директорию, так что вы можете еще раз запустить ее для обработки новых изображений. Если вы хотите »

! Вопрос-победитель



MP3-вещание

В Я хочу использовать мой Linux-компьютер для вещания MP3-файлов. Не снабдите ли меня ключевыми указаниями?

Джеймс Дэвис [James Davis]

О Есть несколько программ для этой цели. Одна из простейших в настройке – Gnump3d (www.gnu.org/software/gnump3d) Стива Кемпа [Steve Kemp]. Когда Вы скачаете и установите ее, все настройки можно отрегулировать в **gnump3d.conf**. Просто следуйте комментариям в этом файле, указывая главные опции, например, расположение MP3-коллекции.

Одна из замечательнейших возможностей Gnump3d – преобразование частоты дискретизации MP3-файлов с помощью Lame на лету. Просто раскомментируйте несколько строк в конфигурационном файле:

```
downsample_high_mp3 = /usr/
bin/lame --mp3input -b 56
$FILENAME -
downsample_medium_mp3 = /
usr/bin/lame --mp3input -b 32
$FILENAME -
downsample_low_mp3 = /usr/
bin/lame --mp3input -b 16
$FILENAME -
```

После этого перезапустите Gnump3d и выберите частоту дискретизации в настройках. Современная машина может перекодировать 2-3 трека одновременно. Если Вы собираетесь вещать достаточно регулярно, можете также установить опцию default_quality. **КК**

в качестве имени файла использовать именно дату получения изображения, то замените **$(date…** на

```
$(date -r ${PIC} +%Y%m%d%H%M%S).png
```

Теперь будет использоваться время последней модификации файла. man-страница по date содержит информацию о множестве возможных опций. Более сложный вариант – читать EXIF-тэги изображения. На это есть множество программ; я предпочитаю *Exiftool* (www.sno.phy.queensu.ca/~phil/exiftool). **НВ**

9 Хочу все сразу!

В Я установил Fedora Core 4, и не знаю, что лучше: Gnome или KDE. Могут ли они быть установлены на один компьютер? Еще, я скачал *K3b*, но не смог установить его. Вы не знаете, почему?

Ян Бирса [Jan Birsa]

О Да, можно иметь более одного рабочего стола на одном компьютере. Снизу от полей ввода имени пользователя и пароля на экране входа в систему, есть меню «Сессия». Оно позволит выбрать, какой рабочий стол загружать. Если Ваша система настроена на загрузку Gnome сразу, по умолчанию, то выберите пункт «Завершить сеанс» в меню «Система». Вы увидите экран входа в систему.

Ясно, что для возможности выбора из этого меню KDE должен быть уже установлен. Скорее всего *K3b* не удалось установить из-за отсутствия библиотек KDE (весь KDE для запуска *K3b* иметь не обязательно). Аналогично, работая в KDE, Вы можете запускать в нем программы для Gnome, если у Вас установлены библиотеки Gnome. **НБ**

10 Прок от /proc

В Когда я выполняю команду *mount*, я вижу файловую систему **/proc**, которой нет на моем диске. Расскажите, пожалуйста, что это и зачем оно здесь?

Даррен Биркетт [Darren Birkett]

О В типичной Linux-системе, выполняя *mount*, Вы должны видеть как минимум две файловых системы, не доступные обычным путем. Первая из них – **/proc**, вторая же отображается как **none on /dev/shm**. Как Вы, может быть, знаете, **/dev/shm** – это файловая система, используемая для управления виртуальной памятью и не присутствующая на локальном жестком диске.

/proc хранит виртуальные файлы, которые являются как бы окном в текущее состояние работающего ядра. Она не занимает места на жестком диске и, следовательно, является виртуальной файловой системой, однако работает и выглядит как обычная дисковая ФС.

Просмотр некоторых файлов в **/proc** может дать хорошую информацию о вашей системе. Так, если вы взглянете на **/proc/meminfo**, то увидите информацию об используемой в вашей системе памяти:

```
# cat /proc/meminfo
MemTotal:      515484 kB
MemFree:        74656 kB
Buffers:       5912 kB
Cached:        352464 kB
SwapCached:         12 kB
Active:     126788 kB
Inactive:    289772 kB
```

Взглянув на эти данные, Вы увидите не только объем памяти в системе (включая раздел подкачки), но и ее текущее состояние с точки зрения свободного места. При новом запуске команды некоторые параметры, скорее всего, изменятся, и это – ключевая особенность **/proc**. Это что-то вроде снимка текущего состояния системы. Более продвинутые пользователи могут временно изменять функциональность ядра, путем редактирования некоторых файлов в **/proc**. Например, для включения маршрутизации IP-пакетов (чтобы система могла функционировать в качестве шлюза, пересылая данные с одного сетевого интерфейса на другой), Вы должны выполнить такую команду:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

Обратите внимание, что это состояние не постоянное и будет сброшено при следующей перезагрузке. Чтобы сделать его постоянным, нужно отредактировать файл **/etc/sysctl.conf**, добавив в него следующую строку:

```
net.ipv4.ip_forward = 1
```

Чтобы узнать больше, прогуляйтесь по **/proc**. Обычным просмотром вы ничего не испортите, а если испортите что-то при редактировании, после перезагрузки все изменения в **/proc** исчезнут. **КК**

11 Потерянный в эфире

В Хотел бы подключиться к широкополосному кабельному Интернету от NTL, но они сказали, что не поддерживают Linux. Значит ли это, что система не работает в Linux, или просто они не могут мне ничего посоветовать? Я не сомневаюсь, что если я подключу кабельный модем к сетевой карте, сеть будет работать, но как потом соединиться с Интернетом? Надо ли использовать *KPPP* с какими-то особенными настройками?

Эдриен Хоррокс [Adrian Horrocks]

О Могу уверить Вас, что с интернетом от NTL в Linux работать можно – я сам это делаю. Надо только соединить сетевой порт модема с сетевой картой вашего компьютера (cross-over тут не нужен) и установить настройки сети в DHCP. KPPP предусмотрен для работы с телефоном, кабельное соединение его не использует.

Включите модем и подождите, пока загорятся индикаторы RDY и SYNC: это означает, что модем соединен с NTL. Теперь можно подключить к модему сетевую карту, и модем назначит Вам IP-адрес, установит настройки шлюза и DNS. Однако, как Вы верно подметили, NTL, да и большинство ADSL-провайдеров, работает с Linux, но поддержку не оказывает. Единственные исключения (по крайней мере в Великобритании) – UK Linux (www.uklinux.net) и The UK Free Software Network (www.ukfsn.org).

Какого бы провайдера Вы ни выбрали, главное – иметь Ethernet-модем. NTL такие предоставляет, но большинство ADSL-провайдеров предлагают USB-модемы. «Бесплатный» модем Вы, конечно, забирайте, но и купите Ethernet-вариант, за 20 евро или около того. **НБ**

12 Семимильными шагами…

В Я новичок в Linux, и, поиграв с некоторыми дистрибутивами (лучший способ научиться чему-то), почувствовал себя в Linux вполне уверенно. Могу пользоваться терминалом и вводить команды, открывать RPM-пакеты и диагностировать систему (если я сталкиваюсь с проблемой – обращаюсь к Интернету). Я бы был вам очень благодарен, если бы вы публиковали руководства, помогающие новичкам быстро освоиться в этой ОС. Я знаю, что дистрибутивов много и очень сложно сделать руководство, которое покрыло бы их все, но я, вероятно, мог бы найти тот, что проще для изучения. Если нет, то отошлите меня, пожалуйста, к руководству, которое я мог бы прочесть на досуге.

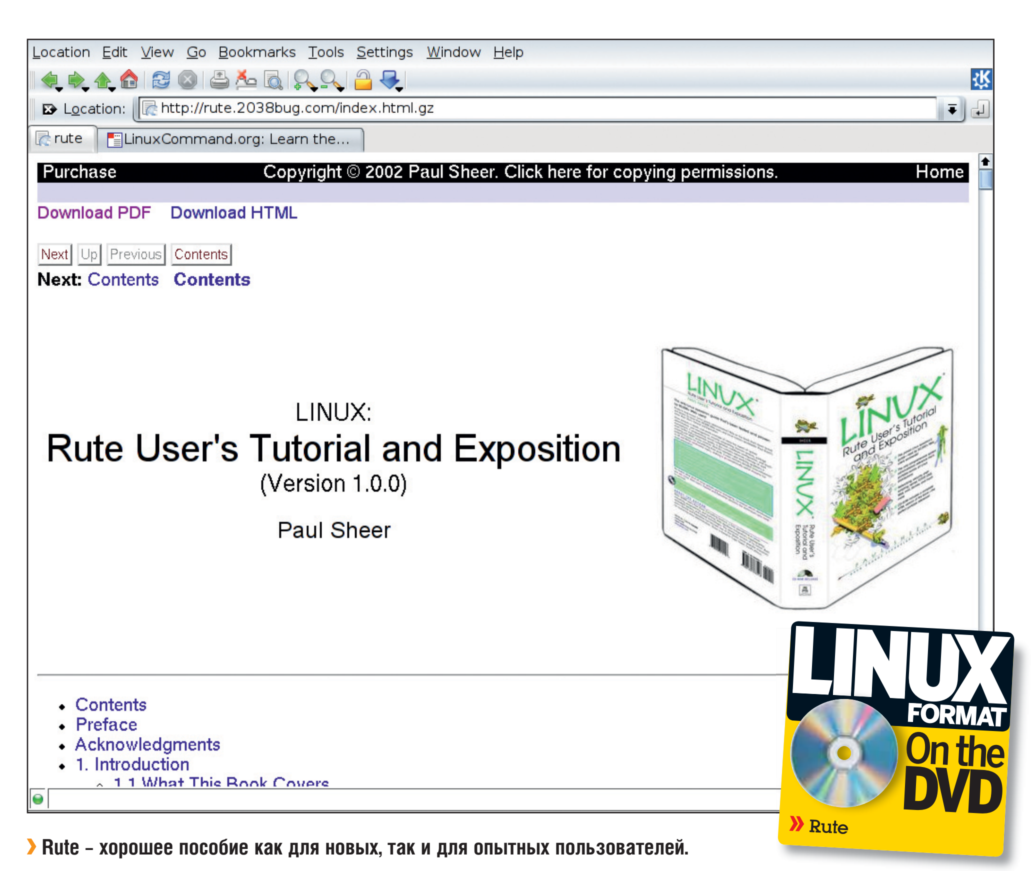
Майкл Квин [Michael Quin]

О Наша серия учебников «Первые шаги» покрывает большую часть того, что Вы хотите. Но лучшие дистрибутивы для

› **Установите сколько угодно рабочих столов и выбирайте между ними при входе в систему.**

изучения Linux частенько менее дружественны для начинающих пользователей. Есть старое высказывание: «Используйте Red Hat, если хотите изучить Red Hat; используйте Slackware, если хотите изучить Linux». В первую категорию также попадают Mandriva и SUSE, во вторую – Arch, Debian и Gentoo. Графические утилиты настройки «защищают» пользователя от углубления в систему, а это тормозит процесс обучения.

Есть множество сайтов, предоставляющих отличную документацию по изучению Linux для пользователей всех уровней. Один из самых популярных ресурсов – Rute, http://rute.2038bug.com. Вы можете читать его в сети, скачать в форматах PDF и HTML, приобрести в бумажном варианте или взять с нашего диска к этому номеру. Одно из лучших руководств по утилитам командной строки Linux лежит на сайте http://linuxcommand.org. MC



› Rute – хорошее пособие как для новых, так и для опытных пользователей.

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
uname -a >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.



LXF HotPicks



В ЭТО РАЗ ТОЛЬКО ДЛЯ ВАС: Bless » Eqonomize » KanjiSaver » KGliv » Pengupop » Rubrica » Sakura » Z-Lock

Адресная книга

Rubrica

Версия 1.0.15 **Сайт** http://rubrica.berlios.de

В прошлом те дни, когда картотеки Rolodex хватало для хранения всей информации о наших контактах. Десять лет назад пошарить в списке постоянных контактов было не сложно, но сегодня мы соприкасаемся с гораздо большим количеством людей, особенно по работе. Многие программы пытались сделать управление контактами разумнее и элегантнее, чем бумажные методы; *Rubrica* принадлежит к данной области. Это адресная книга для Gnome, но так же хорошо будет работать и в других рабочих столах и оконных менеджерах, при наличии соответствующих библиотек Gnome.

Обидно, что разработчики *Rubrica* совершили типичную ошибку: неоправданно разбили программу на отдельные библиотеки. Для ее сборки из исходных текстов вам придется заодно скомпилировать и установить библиотеку *libral*, от того же разработчика. Мы всегда за совместное использование кода во избежание двойной работы, но поскольку, кроме *Rubrica*, эту библиотеку никто не использует (ее нет даже в огромных репозитариях Ubuntu), то, исходя из удобства и простоты для конечного пользователя, следовало включить ее в основной код!

Тем не менее сборка *Rubrica* должна пройти без фокусов, если у вас установлены devel-пакеты *GTK* и Gnome. При первом запуске программа показывает очень приветливый диалог мастера первичной настройки – вы можете выбрать адресную книгу для загрузки по умолчанию и папку, где *Rubrica* должна хранить свои данные, а также назначить скрипт *Nautilus* для экспорта записей в HTML.

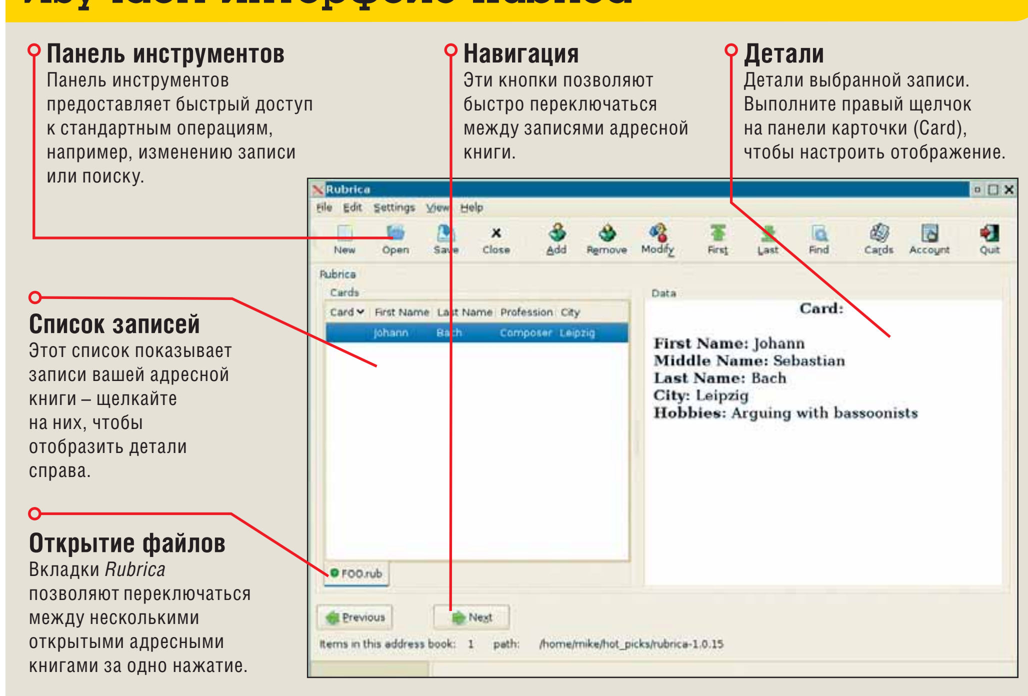


› Редактирование записи: вкладки *Rubrica* позволяют ввести множество данных по каждому контакту.

Благодаря интерфейсу со вкладками, *Rubrica* позволяет иметь несколько открытых адресных книг одновременно: например, одну для работы, а другую с личными контактами. Список контактов отображается в левой панели, и, щелкнув по имени, вы получите справа детальную информацию. Для каждой записи предусмотрено исчерпывающее число деталей: наряду с обычными именем и адресом, можно добавить несколько адресов электронной почты и сайтов, описание должности и фирмы, семейное положение, хобби и многое другое – даже открытый ключ, если человек использует шифрование или цифровую подпись. Еще один подарок – инструмент экспорта в HTML, создающий простые, но очень полезные web-страницы с вашими контактами.

Готовности *Rubrica* к широкому использованию препятствуют несколько мелких ошибок: например, странноватый инструмент поиска и невозможность сохранения положения окна при перезапуске. Но зато здесь есть все функции, предполагаемые в исчерпывающем менеджере адресных книг, и (если не требовать особых причуд) поля для всех данных, необходимых детализированному списку контактов. Мы считаем, что при небольшой доработке *Rubrica* займет важное место в Gnome – так что удачи ее программистам.

Изучаем интерфейс Rubrica



Эмулятор терминала

Sakura

Версия 1.0.9 **Сайт** http://pleyades.net/david/sakura.php

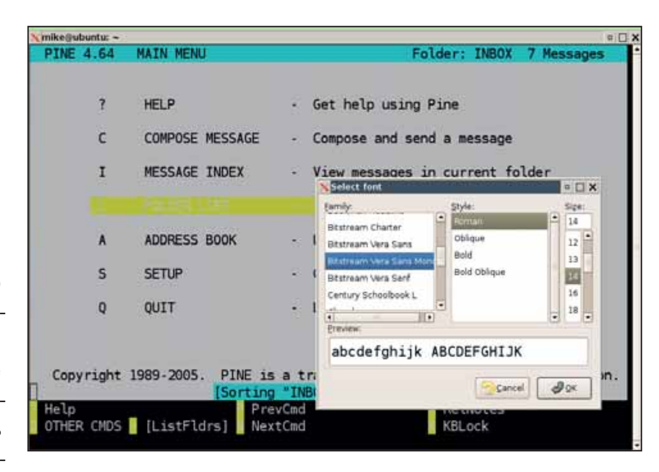
Эмуляторы терминала превратились в постоянно используемый класс программ, и мы ожидаем, что они всегда под рукой, как калькулятор и текстовый редактор, но чаще всего берем тот, что поставляется вместе с нашим дистрибутивом. Однако, как показал обзор в LXF82, у каждого терминала есть много способов превзойти другой, а *Sakura* относительно нов и нацелен на минимум зависимостей и скромные системные требования (насколько это возможно в Gnome).

Базовый компонент *Sakura* – VTE, терминальный движок, используемый в *Gnome Terminal* и *Xfce Terminal*, поэтому для сборки из исходных текстов необходим он и установленные devel-пакеты *GTK*. На данном этапе процесс компиляции довольно необычен: конфигурационный скрипт заменен файлом из системы 'MOBS' с загадочным именем '0'; после распаковки tar-архива сборка осуществляется так:

```
./0
 make
make install
```

(Последняя строка – из-под root.) Затем вводите **Sakura** для запуска программы. По умолчанию *Sakura* напоминает старые эмуляторы Xterm и Rxvt: серо-черная гамма и нет панели инструментов. Однако отличное контекстное меню, вызываемое правым щелчком, поможет изменить цвета и открыть новые вкладки. *Sakura* не обладает ни супер-настраиваемостью, как *Konsole* от KDE, ни удобными профилями, как *Gnome Terminal* – просто работает, как полагается.

Ну, а как с производительностью? Мы сравнили *Sakura* с двумя самыми популярными эмуляторами терминала, запустив Pine и отслеживая использование памяти (RSS): *Sakura* берет 11 МБ ОЗУ – не восторг, но все-таки лучше, чем Gnome Terminal (14 МБ) и Konsole (15 МБ). По части быстродействия, мы засекли время вывода длинного текстового файла (кэшированного в ОЗУ): *Sakura* потребовалось 1.6 секунд, *Gnome Terminal 1.5*, *Konsole 2.1*, а *Xterm* перевалил за 4.4. Опять же, не дьявольски быстрый, но не худший из собратьев, и хорош для любых машин, кроме самых старых.



› В *Sakura* запущен почтовый клиент *Pine*, демонстрирующий работу с ncurses (на переднем плане – диалог настройки шрифта).

> «Sakura – идеальный выбор терминала, если вам нужна функциональность VTE.»

Sakura – идеальный выбор терминала, если вам нужна функциональность VTE-компонента без больших затрат ресурсов и продирания сквозь пугающие бездны опций. Пользователи Gnome, желающие сэкономить системную память, пробуйте!

Инструмент обучения языку

KanjiSaver

Версия 0.9.10 **Сайт** http://rschultz.ath.cx/code.php

Японский язык невероятно тяжел в изучении. Помимо специфического синтаксиса, для чтения японских текстов вы должны освоить аж три вида записи: кандзи, хирагана и катакана. Два последних – слоговые азбуки, поэтому ограничены в числе символов, а канжи происходит от китайского письма, использующего много тысяч иероглифов. И чтоб хотя бы понять японскую газету, нужно знать их не менее 2000. Легко никому не покажется, но благодаря программам вроде *KanjiSaver* изучение языка не превращается в наказание.

KanjiSaver – хранитель экрана для KDE, высвечивающий иероглифы кандзи вместе с их произношением и значением, с целью помочь в пассивном изучении языка: вы не сидите, пытаясь задолбить слова, а видите их каждый раз при запуске хранителя экрана и, вероятно, припоминаете в свободное время. После сборки и установки можно установить программу в качестве хранителя экрана в Центре управления KDE – если не получается, или у вас другой рабочий стол или WM, можете просто запустить **KanjiSaver.kss** из каталога **/src** и видеть его в окне.

После запуска *KanjiSaver* отображает черный экран с кандзи-символом в середине, плюс его произношение (катакана) в верхнем левом углу; в правом нижнем помещается значение на английском. Вот и все – никаких дурацких эффектов или деталей оформления, только иероглиф и его значение. *KanjiSaver* выбирает случайный символ из своего словаря каждые десять секунд, предоставляя вам достаточно времени, чтобы увидеть изображение и (надеемся) запомнить его. В текущем релизе более 1000 символов, этого более чем достаточно, чтобы избежать повторений, и всегда можно добавить новые в файл каталога исходных текстов **/scripts**.



› Невероятно, но факт: японские дети в начальной школе изучают 1006 иероглифов.

> «Вы видите слова при каждом запуске хранителя экрана.»

В HotPicks мы обычно не рассматриваем такие экзотические программы, как *KanjiSaver*, но его потенциал больше, чем просто обучение японскому. В идеале разработчики расширят его другими языками, сделав время простоя наших компьютеров продуктивнее и снизив скуку запоминания слов из учебника. В любом случае, он интереснее, чем обычные прямоугольники или звездное небо, которые мы видим на 99.999% компьютеров...

Менеджер финансов

Eqonomize

Версия 0.2 **Сайт** http://eqonomize.sourceforge.net

Быть мультимиллионером, должно быть, здорово. Можно запускать Linux на машине-монстре со 128 SGI-процессорами, оплачивать написание новых версий ваших любимых игр и швыряться деньгами со щенячьей энергией. Увы, большинство из нас ведет строгий учет личных средств – приход, расход, и куда девалась зарплата. Поэтому на свете множество приложений с открытым кодом для управления финансами, и недавно к ним прибавился *Eqonomize*! (да, восклицательный знак входит в название, но для простоты мы его далее опустим).

Детище Никласа Кнутссона [Niklas Knutsson], *Eqonomize* фокусируется на простоте использования и эффективности для «малой домашней экономики». Он не предназначен для управления расходами предприятий, но домашним пользователям предоставляет неплохой набор функций, даже на столь ранней стадии разработки. Основная цель программы –слежение за транзакциями, планированием бюджета и ценными бумагами (акциями, облигациями и т.п.), а также генерация отчетов для быстрой оценки вашего финансового состояния.

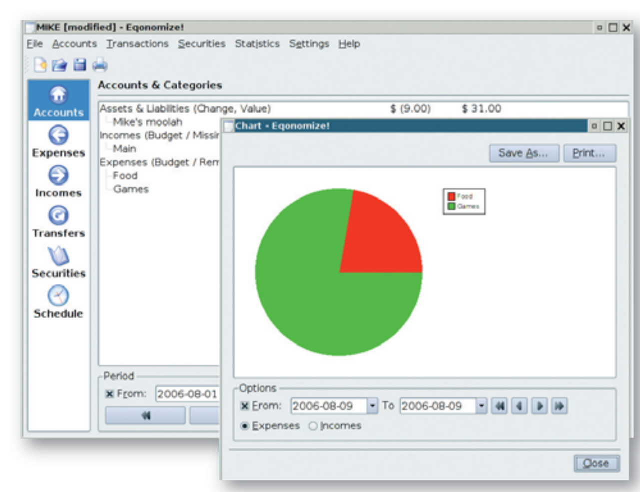
Eqonomize – программа для KDE; для сборки из исходных текстов вам потребуются установленные devel-библиотеки KDE версии не старее 3.2, а также *Qt 3.x*. В большинстве дистрибутивов можно установить их, поискав в репозитариях **kde-devel**, **kdelibs-dev** или схожие имена пакетов (в Ubuntu, например, понадобятся **kde-devel** и **libqt3-mt-dev**). Распакуйте **Eqonomize-0.2.tar.gz** с диска-приложения, перейдите в получившийся каталог и введите стандартное **./configure**, **make** и **make install** (последняя команда – от имени root).

Теперь просто запускайте программу, набрав в командной строке **Eqonomize**, или добавьте исполняемый файл на ваш рабочий стол. Хотя *Eqonomize* лучше всего подходит для KDE, он без проблем запускается и в других рабочих столах и оконных менеджерах.

Главное окно *Eqonomize* демонстрирует хорошо продуманное эргономичное размещение элементов. Внизу слева вы найдете шесть иконок для переключения между различными видами ваших финансов: все ваши счета, расходы, доходы, переводы, ценные бумаги и запланированные сделки. Щелкните на пунктах меню File > New – и начнете с чистого листа, где можно добавлять транзакции (*подробнее см. внизу*).

> «Кроме документации, здесь вряд ли чего-то не хватает.»

Отметим, что *Eqonomize* поддерживает пока только знак доллара, но для работы в евро или рублях разницы никакой: все эти валюты десятичные, так что отличие сугубо косметическое. После настройки учетной записи можно создать категории для входящих и исходящих платежей – счета, еда, квартплата и т.д. Каждая запись может использовать опцию повторения, это позволяет точно определить, как часто деньги приходят/уходят – например, в первый выходной каждого месяца.



› *Eqonomize* щеголяет круговыми диаграммами, прекрасно иллюстрирующими распределение расходов.

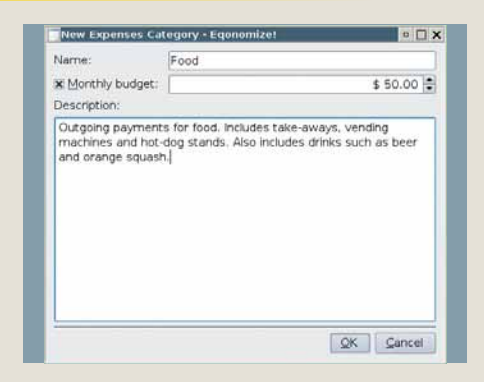
Затем вы можете просмотреть сроки платежей в вашем расписании и даже секторную диаграмму составляющих ваших расходов. *Eqonomize* хранит данные в XML-формате, и их можно экспортировать в HTML или CSV – последний идеален для редактирования в электронной таблице.

Хотя *Eqonomize* едва достиг версии 0.2, для обычного домашнего пользователя здесь вряд ли чего-то не хватает, разве что документации маловато. Он легко осваивается, не проявляет нестабильности и прекрасно оснащен функциями импорта/экспорта. Плюс нас восхитил уровень детализации – в частности, при управлении периодическими транзакциями и вкладами. Ждем не дождемся первого полного релиза: Никлас, лишь знак рубля насущного дай нам – и вперед!

Шаг за шагом: Создание учетной записи

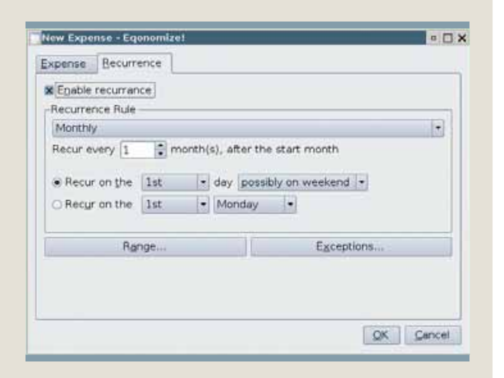
» Создание нового файла

Щелкните File > New в меню для запуска *Eqonomize* с пустыми настройками, и вы увидите это окно учетной записи.



» Создание категорий

Щелкните на Accounts > New Income Category [Новый вид дохода] или New Expense Category [Новый вид расхода], чтобы начать организацию ваших платежей (еда, счета и т.д.).



» Добавление платежей

Теперь щелкните на иконках Expenses [Расходы] или Incomes [Доходы] и добавьте вашу транзакцию. По завершении щелкните Accounts – увидите свой статус.

HotGames Развлекательные приложения

Игра-головоломка

Pengupop

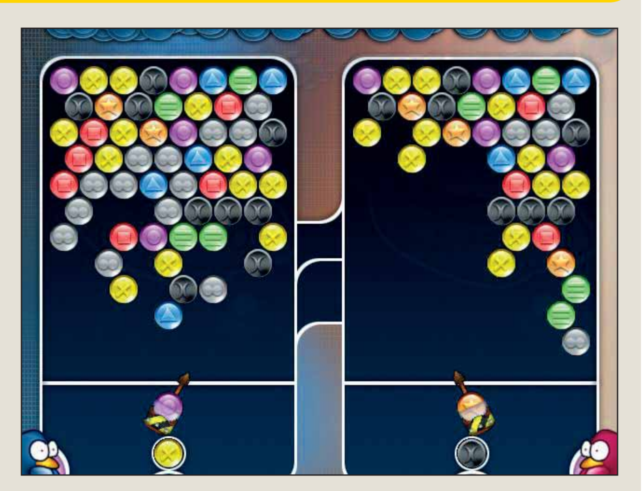
Версия n/a **Сайт** www.junoplay.com/pengupop

Быстро поднимите руки: кто играл во *Frozen Bubble*? Если вы хоть сколько-нибудь проработали в Linux, то вряд ли прошли мимо: это одна из самых популярных и знаменитых игр с открытым кодом, объединяющая приятный вид с классической игровой идеей. *Frozen Bubble* – реинкарнация классического *Puzzle Bobble* (также известного как Bust-a-Move), и *Pengupop* переняла ее игровую механику, добавив пару наворотов от себя.

Pengupop – головоломка, в которой вы стреляете шарами в верх экрана и объединяете их с шарами того же цвета. Объедините три вместе, и они исчезнут, возможно, прихватив с собой некоторые из нижележащих. На словах это объяснить трудно, но гляньте на снимок экрана, и вы поймете идею. *Pengupop* – сетевая многопользовательская игра: вы сражаетесь с другими игроками со всего мира, спихивая шары в их игровые области, а свою очищая.

Слава разработчикам, что они избежали утомительных зависимостей и хитростей установки; *Pengupop* поставляется в виде отдельного исполняемого файла, содержащего все необходимые данные. Понадобится только SDL (стандартно присутствует почти во всех дистрибутивах), так что можно просто запустить **Pengupop-linux-ia32.bin** файл и приступить.

И что еще лучше, *Pengupop* не утомляет настройкой игрового сервера или IP-адреса – вы просто запускаете игру, а *Pengupop* сам находит ожидающего вас противника. Делает он это с легкостью невероятной, и если вы устали от мороки с установлением соединения, то это станет для вас глотком свежего воздуха. Игра еще не слишком широко известна, но сетевого соперника мы находили всегда.



› Как и в *Tetris* для двух игроков, завершение строки засыпает ужасными обломками игровую область оппонента.

В режиме игры *Pengupop* соответствует ожиданиям: игровая механика и обнаружение столкновений кодированы хорошо, смотрится круто, а выстрелы сопровождаются достойными звуковыми эффектами. Как и в *Tetris*, ошибиться, следуя правилам, невозможно, если не добавлять бессмысленных трюков, и мы рады, что программисты *Pengupop* придерживались испытанной концепции игры. Если вам нравятся игры Puzzle Bobble и вы желаете попробовать сыграть с другими игроками по сети, то это почти совершенство.

Стрелялка

Z-Lock

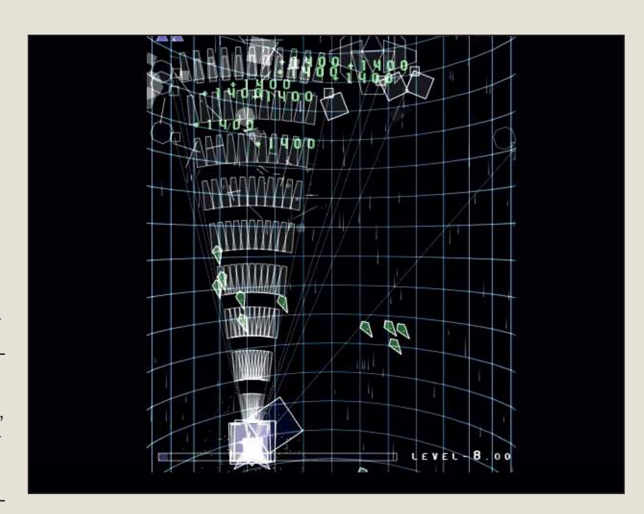
Версия 0.1 **Сайт** www.emhsoft.net/zlock

Мы в редакции LXF ценим умилительно плохие переводы [а уж как их ценим мы здесь, в российской редакции! – прим. ред.], поэтому полюбили вступление к *Z-Lock*: «Будет сильнее если целиться. Теперь оно закреплено ко мне!» *Zero Wing-тастика.*

Эта причудливая японская стрелялка, означенная в файле **readme_e.txt** как 'Hello World Project', представляет собой традиционный 2D-бластер, дополненный необычными игровыми идеями, а также суперплавной графикой; отметим, что при запуске вы можете намучиться с зависимостями. Особо везучие могут просто распаковать архив и запустить **Z-Lock** в получившемся каталоге, вообще без установки. Если игра не работает, то для сборки следует установить компилятор языка D.

Идея *Z-Lock* довольно проста: чем больше вражеских кораблей попадает в ваш прицел, тем сильнее ваша огневая мощь. Так что пока боевые действия на экране идут вяло, вы лишь изредка постреливаете мелкими зарядами, но стоит появиться банде плохих парней, ваше оружие заряжается, и вы можете учинить буйство. Это создает восхитительный игровой процесс: иногда выгоднее какое-то время стоять неподвижно, позволяя вражеским кораблям попасть на мушку, но нужно и не забывать уклоняться от огня противника.

Z-Lock имеет четыре игровых режима: обычный, описанный выше; режим, когда можно стрелять, только если враг сам держит вас в прицеле; атака на счет – набрать максимум очков за три минуты; и атака на время – как можно скорее набрать миллион очков. Управление во всех режимах одинаково: передвижение при помощи клавиш управления курсором и клавиша Z для выстрела.



› Может, это и выглядит свалкой, но в действии *Z-Lock* просто великолепна.

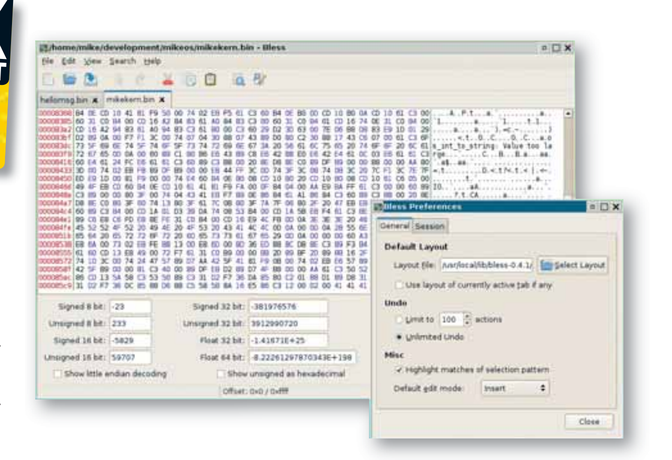
Z-Lock заслуживает похвалы за новый подход к старому жанру платформенных стрелялок, еще усиленный ультрасглаженной анимацией. Игра использует полигоны вместо растровых спрайтов – для врагов, плавно кружащихся по всему экрану – и псевдо-3D эффекты для фона. Прибавьте энергичную музыку в стиле *Thunderforce 4* – и получится минималистичный стиль *Z-Lock*. Даже удивительно, что мы дошли так далеко в этом обзоре и не сказали: «Некто настроить мы бомба!»

Шестнадцатиричный редактор

Bless

LINUX FORMAT Hottest Pick

Версия 0.4.1 **Сайт** http://home.gna.org/bless



› **Редактирование двоичного ядра MikeOS в *Bless* – обратите внимание на многодокументный интерфейс со вкладками.**

Шестнадцатиричные редакторы стары как мир, но они все еще помогают в выполнении важных задач: взламываете ли вы файловые форматы, изучаете ли заголовки изображений или просто копаетесь в исполняемом файле, без такого редактора не обойтись. В противоположность текстовому редактору, работающему только с читаемыми символами, шестнадцатиричный позволяет изменять любые байты файла, в шестнадцатиричном (основание системы счисления 16) формате. Как и следовало ожидать от программеролюбивой системы типа Linux, здесь имеются триллионы редакторов для выполнения этой задачи. *Bless* намерен влезть на вершину этой кучи, благодаря внятному интерфейсу и вагону всякой статистики.

Поскольку *Bless* написан на *C#*, для компиляции и запуска потребуется связка Mono и *GTK#*. Небольшой совет: в зависимости от установленных библиотек, может оказаться, что интерфейс *Bless* неработоспособен из-за размера шрифта. Если после компиляции вы не можете прочесть надписи в окне приложения, отредактируйте строку 47 в **src/gui/drawers/Drawer.cs**, изменив ее на

```
FontName="Courier 8";
```

После этого перекомпилируйте и запустите – отображение исправится. Главное окно *Bless* позаимствовало расположение элементов у превосходного *KHexedit*, эквивалента от KDE, со списком смещений слева, шестнадцатиричными числами посередине и ASCII-представлением справа, помогающим обнаруживать строки в исполняемых файлах; а чередование синего и черного цвета шестнадцатиричных цифр не дает вашим глазам увязнуть в мельтешащей информации.

Можно использовать поиск по файлу (в шестнадцатиричном, двоичном, восьмеричном и текстовом форматах), переходить по смещению и настраивать внешний вид. *Bless* предоставляет массу сведений о каждом байте, слове или двойном слове, например, значение в виде десятичного знакового и беззнакового числа, или числа с плавающей точкой; вдобавок можно переключать порядок следования разрядов в представлении чисел.

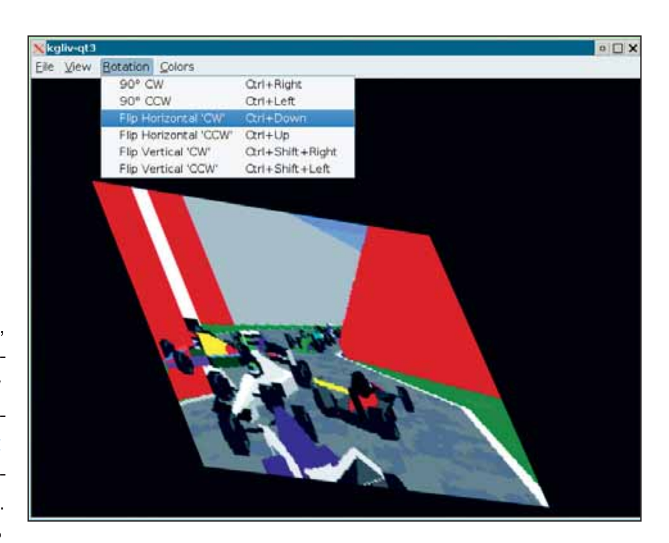
Шестнадцатиричные числа и ASCII-текст можно вставлять прямо в файл, либо добавкой к содержимому, либо, после нажатия клавиши Insert, с перезаписью исходного.

Исключая неувязки со шрифтами, *Bless* – солидный и умелый шестнадцатиричный редактор, достойный включения в арсенал любого программиста, использующего Gnome. Важнее всего то, что он предоставляет поклонникам *GTK* нечто подобное *KHexedit* от KDE, и если вы ищете дружелюбный и информативный инструмент хакера, *Bless* должен стать первым кандидатом.

Просмотровщик изображений

KGliv

Версия 0.7 **Сайт** www.kde-apps.org/content/show.php?content=40392



› **OpenGL-основа *KGliv* помогает крутить, поворачивать и отражать изображения.**

По последним подсчетам, просмотровщиков изображений под Linux больше, чем торговых автоматов в Японии, и может показаться странным. что мы уделяем полстраницы еще одному из них. Так бы оно и было, не имей *KGliv* особенностей, выделяющих его на фоне всяких *Eye of Gnome* – в первую очередь, впечатляющего использования 3D-технологии OpenGL. На вид *KGliv* предоставляет обычные функции поворота и отражения; но делает это стильно.

Как следует из названия, *KGliv* – инструмент KDE, так что для сборки из исходных текстов потребуются заголовочные файлы и библиотеки KDE. Распакуйте tar-архив и введите **make** в получившемся каталоге для компиляции. Однако полезно проверить прилагающийся исполняемый файл в каталоге **bin**, который может просто заработать в вашем дистрибутиве – выполните **KGliv-qt3 <имя файла>**, чтобы проверить, запускается ли он.

После запуска *KGliv* не показывает ничего особенного, просто отображает указанное изображение в виде миниатюры посреди экрана. Но поэкспериментируйте с мышью, и вы узнаете почерк OpenGL-основы: например, придержите клавишу Ctrl и левую кнопку мыши, затем подвигайте мышь, и изображение плавно повернется. Используйте Ctrl+Shift и левую кнопку мыши для поворота изображения вокруг оси X, и Ctrl+Alt для оси Y. Нажимая клавиши + и –, вы можете изменять масштаб отображения картинки.

Перейдите в меню Rotation (Вращение), чтобы увидеть в работе более искусные эффекты, например, функцию отражения, вращающую изображение в окне вокруг его оси. Каждая из этих операций вовсю использует OpenGL; вместо замены изображения на конечный результат, вы видите происходящие преобразования как полигональную анимацию. Пусть это скорее стиль, чем необходимость, но работать с ним как-то приятнее, чем с обычным просмотрщиком изображений.

> «Обычный на вид поворот или отражение Kgliv делает стильно.»

К сожалению, *KGliv* подвержен ошибкам: например, при переключении в полноэкранный режим изображение исчезает. Но это нетрудно исправить, а в остальном *KGliv* – отличный, компактный просмотрщик изображений, и прекрасное дополнение к *Xgl* и компании при демонстрации Linux друзьям и коллегам. В конце концов, симпатяшки всегда привлекают!

Клиент BitTorrent

KTorrent

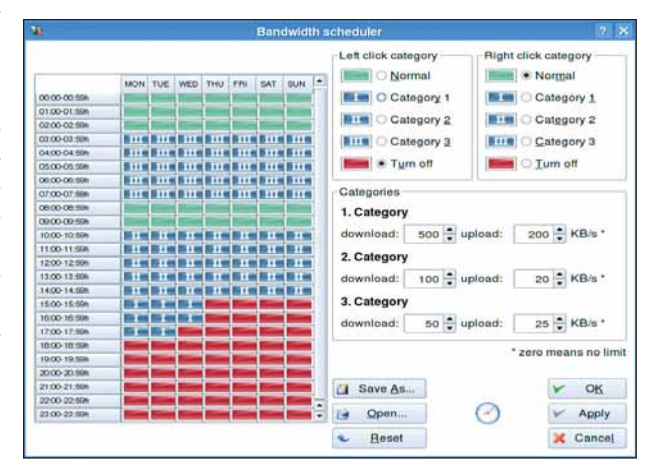
Версия 2.0 **Сайт** http://ktorrent.org

LINUX FORMAT HotPicks повторный визит

Едва минул год, как мы рассматривали *KTorrent* в HotPicks (LXF74), но разработка этого клиента BitTorrent идет с головокружительной скоростью. В нашем обзоре версии 1.1 мы описали его как небольшое компактное приложение с достаточным числом функций для удовлетворения всех, кроме самых привередливых пользователей. Чтобы собрать его из исходных текстов, вам понадобится KDE версии не менее 3.4 и установленная библиотека GNU mp. Раз вы это читаете, значит, есть уже и бинарный пакет для вашего дистрибутива.

KTorrent предоставляет уйму данных о статусе ваших загрузок и допускает расширение, через прекрасную систему встраиваемых модулей. Но лучшее нововведение версии 2.0 – поддержка DHT (distributed hash tables – распределенные хэш-таблицы), дающая доступ к потокам, не имеющим трекера. Другим большим добавлением является диспетчеризация трафика: можно устанавливать полосу пропускания для различных времен суток, чтобы не прерывать работу в течение дня. Теперь поддерживается шифрование, а средства сканирования каталогов изменены так, что программа автоматически загружает torrent-файлы при запуске.

Исправлено множество ошибок, что изменило статус *KTorrent* от небольшого *BT*-клиента до реального соперника могучему *Azureus*. Если вы – пользователь KDE и все еще возитесь с утилитами командной строки BitTorrent, попробуйте *KTorrent*.



› Новая функция диспетчеризации трафика защитит вашу сеть от полного отказа.

Операционная система

MenuetOS

Версия 0.82 / 0.48b **Сайт** www.menuetos.net

LINUX FORMAT HotPicks повторный визит

Если вам попадется в руки LXF62, просмотрите раздел HotPicks с нашим первым обзором MenuetOS, операционной системы с открытым кодом, целиком созданной на Ассемблере (по сути, в машинном коде). Этот изумительный проект вырос в готовую к использованию ОС с графическим интерфейсом и сетевым стеком, благодаря небольшой, но преданной команде профессионалов Ассемблера, придавших невероятную функциональность образу одной дискеты, работая с частичной занятостью.

MenuetOS теперь сфокусировалась на 64-битной версии, хотя и над оригинальной 32-битной инкарнацией работа продолжается и достигла уже версии 0.82. С момента выпуска в июне 2005 г. 64-битной MenuetOS версии 0.01, проект мчался семимильными шагами, портируя код из первоначальной версии и добавляя поддержку TCP/IP, звуковые драйверы, простой почтовый клиент и многозадачность. Приняты меры по обеспечению совместимости с приложениями 32-битной MenuetOS – ход хороший, поскольку сохранит много тяжелого труда, вложенного разработчиками в первоначальную версию. MenuetOS не скоро бросит вызов Vista, Linux или OS X – а может, и останется не более чем диковиной – но даже если вы не интересуетесь программированием, то подивитесь на достижения разработчиков. Как тут не удивляться, почему типичные приложения, используемые нами день ото дня, все распухают и замедляются (хе-хе, *OpenOffice.org*, хе-хе). LXF




› MenuetOS 64 с сессией разработки на Ассемблере во всей красе.

Также выпущены

Новые и обновленные приложения, также заслуживающие внимания...

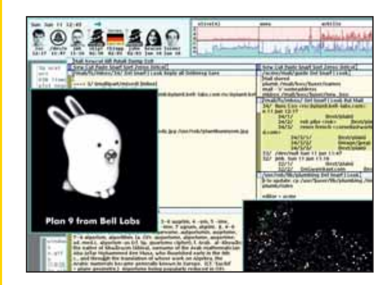
Mahogany 0.67 Почтовый клиент GTK http://mahogany.sourceforge.net

Warrior 0.95.3 Java-браузер http://html.xamjwg.org/browser.jsp

Allegro 4.3.0 Библиотека для разработки игр http://alleg.sourceforge.net

Mednafen 0.6.3 Эмулятор множества систем (GB, NES, PC Engine и т.д.). http://mednafen.com

Plan 9 20060809 ОС типа UNIX http://plan9.bell-labs.com/plan9



› Plan 9: бойтесь эмблемы кролика!

Uncrustify 0.0.24 Расчистка исходных текстов http://uncrustify.sourceforge.net

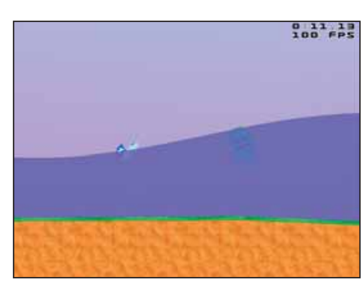
FreeVMS 0.2.12 Клон VMS с открым кодом http://tinyurl.com/6q5gz

PodBrowser 0.10 Браузер документации Perl http://jodrell.net/projects/podbrowser

LedaX 0.02 проектирование электронных схем для KDE www.yaeda.org

Dkop 0.10 Инструмент для полного/инкрементного резервирования на DVD http://kornelix.squarespace.com/dkop

Bloboats 1.0.1 Игра: гонки на бешеных лодках http://bloboats.dy.fi



› Bloboats: был бы простецким, кабы не шелково-гладкая анимация скал.

CrossFTP 1.00 FTP-сервер http://crossftp.googlepages.com

Aria2 0.7.0 Менеджер загрузок для командной строки http://aria2.sourceforge.net

Goby 0.2.1 ACDSee-подобный просмотровщик изображений http://goby.sourceforge.net

LXF DVD 85

Готовьтесь встречать Linux для предприятий со SLED 10!



Майк Сондерс
Любовно составляет диски Linux Format и поддерживает сайт **www.linuxformat.co.uk**.

Linux значит бизнес

Мы буквально в восторге от нашего основного дистрибутива этого месяца: SLED –действительно впечатляющая разработка, объединившая лучшие качества SUSE и эдакий корпоративный лоск. SLED нацелен на бизнес, но это не помешает вам попробовать его и дома – в нем есть все необходимое для настольной системы Linux, с хорошо протестированными основными компонентами. А если вы намучились, пытаясь заставить *Xgl* работать с другим дистрибутивом, то SLED избавит вас от страданий, поскольку классные 3D-эффекты полностью интегрированы и достигаются простым щелчком мыши.

Еще одна изюминка нашего DVD – ReactOS 0.3.0. Возможно, вы помните обзор этого потрясающего проекта в LXF80 HotPicks: разработчики нацелились на создание ОС с открытым кодом, совместимой с программами и драйверами Windows. Не «надстройки» типа *Wine*, а полноценной операционной системы! Успехи, достигнутые за последние месяцы, потрясают: давно ли ReactOS мог осилить только базовые программы, например, *Notepad*, а сегодня ОС в состоянии работать с такими полновесными приложениями, как старые версии *Photoshop, MS Office* и мощный *Unreal Tournament* – совместимость улучшается день ото дня. Хотите узнать больше? Переверните страницу!

На диске есть еще много интересного: документация *Rute*, новые инструменты разработки, парочка аудио программ и несколько игр для приятного досуга. Читайте дальше и наслаждайтесь! А если у вас есть рекомендации по поводу диска, скиньте их мне на почту: mike.saunders@futurenet.co.uk

Краткое содержание DVD

Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

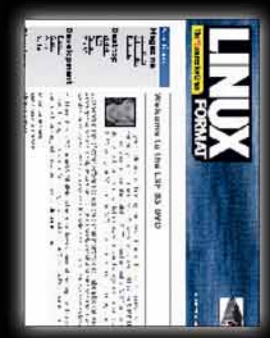
имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

LINUX FORMAT 8 GIGABYTE DVD

Плюс...
Glade 3.0: Приложения Gnome – это просто!
PenguinTV: Легкий путь к подкастам и видеоблогам
Blender 2.42a: Мощная открытая система 3D-моделирования
ReactOS 0.3.0: Открытый клон Microsoft Windows
Wormux: Старые добрые Worms – под Linux!





Диск Linux Format: способ применения

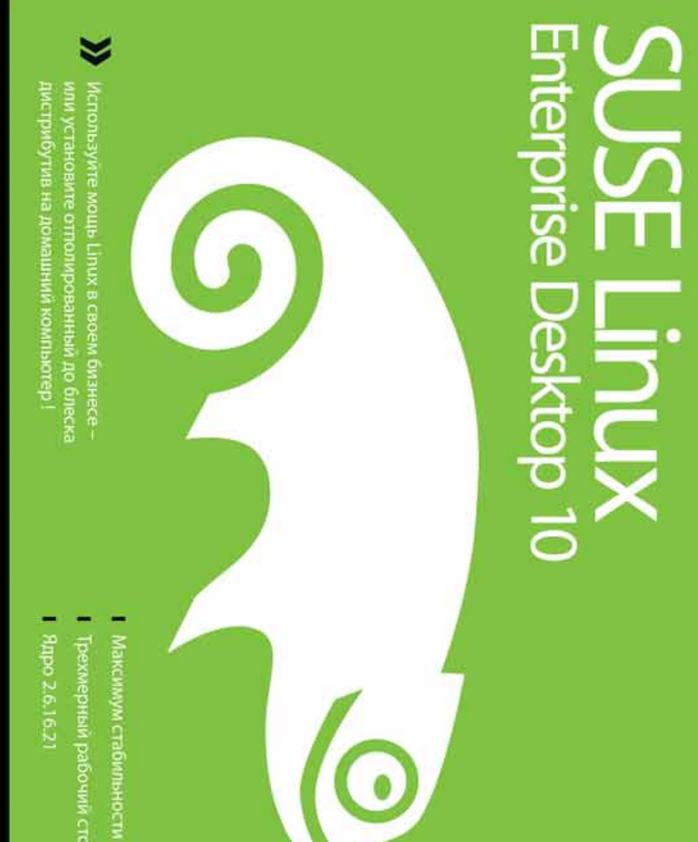
1. Вставьте диск в привод DVD-ROM и откройте index.html
2. Щелкайте по ссылкам слева, чтобы открыть различные категории
3. Программы поставляются в виде исходных текстов и, по возможности, двоичных пакетов

ноябрь 2006

LXFDVD85



SUSE Linux Enterprise Desktop 10



Используйте мощь Linux в своем бизнесе – или установите отполированный до блеска дистрибутив на домашний компьютер!

- Максимум стабильности
- Трехмерный рабочий стол
- Ядро 2.6.16.21







Содержание LINUX FORMAT DVD Ноябрь 2006

Сторона 1

Рабочий стол
Filelight – приложение для создания графика используемого места на диске
K3b – программа для записи компакт-дисков
Sbackup – простая в использовании утилита для резервного копирования
Tagore – персональный вики для среды Gnome
Zhu3D – трехмерный просмотрщик графиков функций

Разработка
Bazaar – децентрализованная система управления версиями
Glade – утилита для построения пользовательских интерфейсов
Olive – графический интерфейс к Bazaar
Scite – текстовый редактор на базе Scintilla

Дистрибутивы
PCLinuxOS MiniMe – облегченная версия PCLinux
ReactOS – свободный клон Windows NT
SLED – корпоративная версия SUSE от Novell

Игры
Bloboats – гоночная игра в духе ElastoMania и Xmoto
Fish Supper – аркадная игра, клон Frogger
Wormux – пошаговая стратегия в стиле Worms

Графика
Blender – открытый 3D-редактор
Cinelerra – нелинейный видеоредактор
gkII – генератор фракталов
Inkscape – свободный векторный редактор
Scribus – приложение для верстки

Справка
RUTE – книга по администрированию Linux

Горячие новинки
Bless – hex-редактор
Eqonomize! - программа учета финансов
KanjiSaver – хранитель экрана
Kgliv – просмотрщик изображений
Ktorrent – bittorrent-клиент
MenuetOS – любительская ОС, полностью написанная на ассемблере
Pengupop – аркадная игра
Rubrica – адресная книга для Gnome
Sakura – эмулятор терминала
Z-lock – аркадный шутер

Интернет
Azureus – bittorrent-клиент с богатой функциональностью
Feedoocha – RSS-генератор
PenguinTV – RSS-агрегатор для работы с мультимедийным контентом

Звук
Amarok – самый функциональный из существующих аудиоплеер
gmusicbrowser – аудиоплеер с менеджером коллекции
Traverso – мультитрековый аудиоредактор

Система
Bastille – программа для усиления безопасности операционной системы
Cairo – библиотека вывода двумерной графики
Krdm – менеджер сеансов удаленного доступа
Squashfs – файловая система с функцией сжатия

Сервер
PostgreSQL – продвинутая СУБД

Журнал
Cinelerra - Мультфильм, смонтированный в Cinelerra
Compiz - Файлы из учебника Compiz
KDE - Файлы из учебника KDE
Linux like Mac - Программы, упомянутые в статье «Многоликий Linux»
Scribus - Журнал EMULATORS MACHINE 6, сверстанный в Scribus
Unix API - Файлы из учебника «Программирование для Unix
Сравнение - Набор Java IDE

Сторона 2

SUSE Linux PackMan Repository – репозиторий дополнительных пакетов к SUSE Linux 10.1.

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен на Уральском электронном заводе., 620066, Россия, г. Екатеринбург, ул. Комвузовская 17-203, Лицензия МПТР России ВАФ № 77-15

Поставляется вместе с журналом LINUXFORMAT номер 11(84) Ноябрь 2006



СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

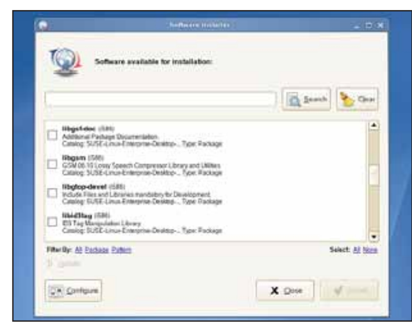
Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. **LXF**

Дистрибутив Linux

SLED 10

Без сомнения, вы жаждете испытать SUSE Linux Enterprise Desktop 10 – это фантастическая основа SUSE, которую мы знаем и любим, плюс установленный по умолчанию *Xgl*, опциональная долгосрочная поддержка и миллионы человеко-часов тестирования – короче, самый отшлифованный и стабильный из текущих дистрибутивов. Если ваша фирма использует Windows и надо убедить начальство, что Linux – прекрасная ОС для настольных компьютеров уровня предприятия, именно этот дистрибутив способен вам помочь. А если вы – домашний пользователь и ищете дистрибутив с долгим жизненным циклом, который не будет обновляться каждые 15 секунд, вы и тут не ошибетесь. Нет нужды много говорить о SLED; откройте стр. 30 – там есть вся необходимая информация!

Для установки SLED 10, просто загрузите ПК с нашего DVD и выберите Installation в меню загрузки. (Убедитесь, что ваш BIOS настроен на загрузку с DVD, а не с жесткого диска.) Компьютер 500 МГц CPU и 256 МБ RAM будет притормаживать – лучше иметь 512 МБ и 1.5 ГГц. Вам потребуется не менее 3 ГБ на жестком диске, чтобы впихнуть на него все прелести! Программа установки SLED очень проста в использовании; если у вас возникнут проблемы, нажмите кнопку Help и следуйте руководству по установке – и все будет хорошо. Не забудьте, что для компьютеров только с CD-приводом можно создать CD-версию SLED при помощи системы Jigdo, сделав ISO-образы на компьютере с DVD-приводом. Подробности – в файле **index.html** на диске.



› **Добавлять новые программы с помощью диалогового окна этого Software Installer удивительно просто.**

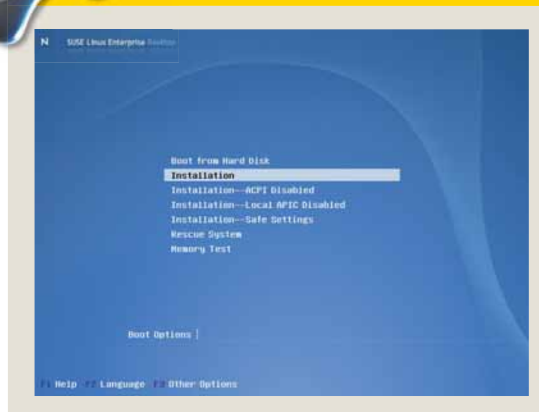


› **SLED по умолчанию использует Gnome с новыми меню и панелями управления. Читайте об этом на стр. 30.**

Поддержка

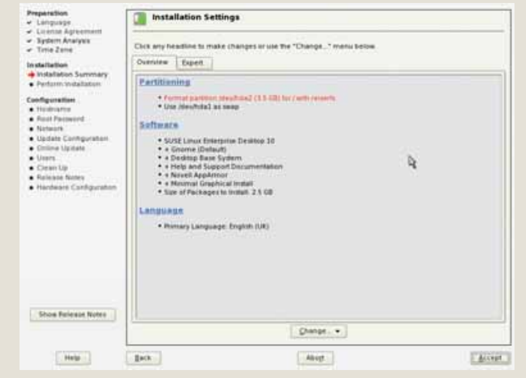
Чтобы в течение 60 дней получать обновление пакетов от Novell для SLED 10, посетите http://download.novell.com, щелкните на SUSE Linux Enterprise Desktop 10, затем – на кнопку Get Activation Code (Получить код активации) внизу. Обновление пакетов в течение года стоит $50. И – как всегда в Open Source – под рукой есть сообщество пользователей, которое всегда готово помочь: если нужен совет, посетите www.suseforums.net и наши форумы на www.linuxformat.co.uk/forums.

Шаг за шагом: Установка SLED 10



1 Загрузка

Загрузите ваш ПК с DVD и выберите Installation (Установка) в этом меню. Если возникнут проблемы с загрузкой программы установки, используйте опцию Safe Settings.



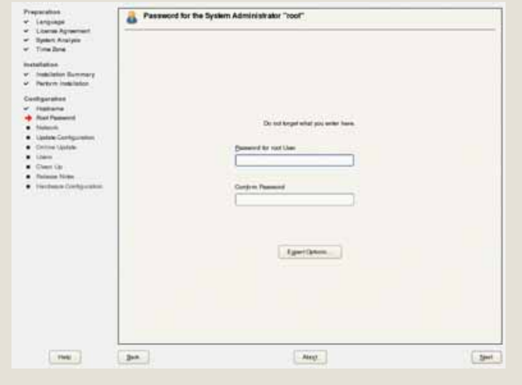
2 Запуск

Выберите нужный язык, прочитайте лицензионное соглашение и переходите на экран анализа системы (System Analysis). Здесь можно изменить набор программ по умолчанию или схему разбиения диска.



3 Установка

Если вы решили переразбить диск, то сначала создайте резервные копии данных! После выделения необходимого места для SLED начнется копирование файлов. Это может занять от 10 до 60 минут.



4 Завершение

После установки компьютер перезагрузится (извлеките DVD) и произведет окончательную настройку. Теперь можете входить в систему!

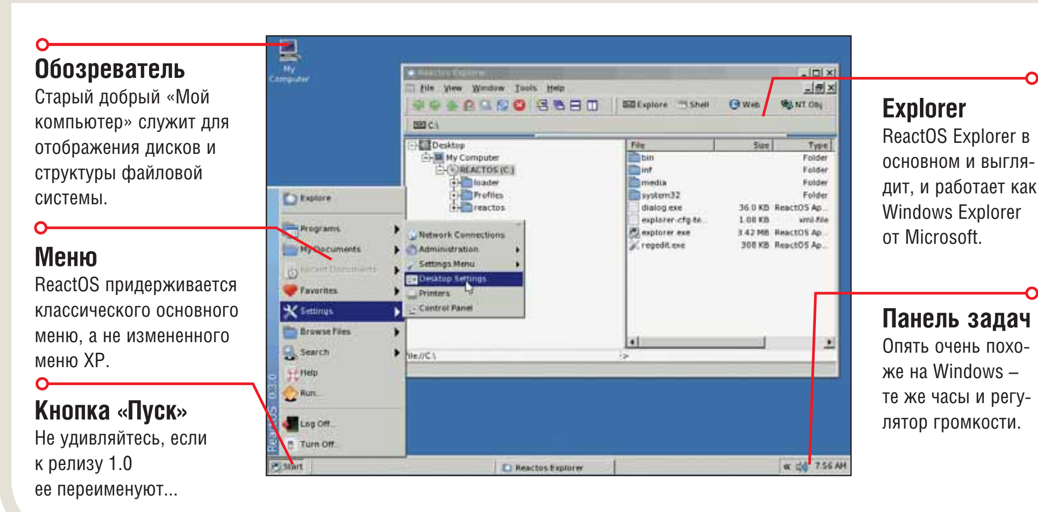
Операционная система

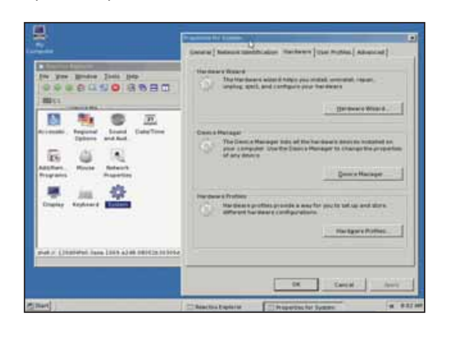
ReactOS 0.3.0

От фактов не отмахнешься: хотя Linux бесплатен как пиво и свободен как слово, стабилен и не боится вирусов, кое-кому все еще нужен Windows. А нельзя ли иметь Windows-подобную ОС и все-таки наслаждаться свободой и надежностью открытого кода? Ответом может стать ReactOS – это свободная ОС, способная [в теории, – прим. ред.] работать с программами и драйверами Windows. Она имитирует элементы графического интерфейса пользователя и файловую систему Windows, упрощая переход с Windows на ReactOS до предела.

Мы уже рассматривали ReactOS в разделе HotPicks в LXF80; хотя это версия всего 0.3.0, она уже справляется со многими небольшими программами Windows, и совместимость с другими растет. Например, она умеет работать с Windows-версиями *Firefox* и *Abiword*, с *Internet Explorer 7* и *Photoshop 5*. ReactOS пока не готова для ежедневного использования: многовато ошибок, да и проблемы совместимости полностью не решены; но попробовать ее очень заманчиво, и вы можете прокрутить ее прямо из раздела **Distros** нашего DVD.

Знакомство с рабочим столом ReactOS



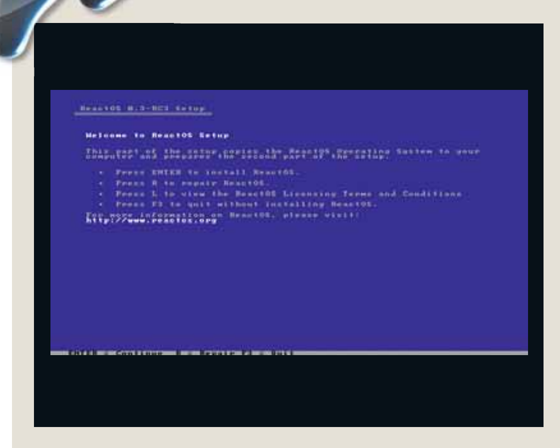


› Для упрощения перехода, панели управления смоделированы по образцу Windows.

Есть два способа запустить ReactOS: либо записать **reactos0.3.0-REL-live CD** и загрузиться с него (в ОЗУ), либо записать установочный CD и загрузиться с него с установкой ReactOS на жесткий диск. Учитывая, что ReactOS все же находится в процессе разработки, мы рекомендуем придерживаться варианта с Live CD, если у вас нет запасной машины и вы не хотите рисковать другими разделами вашего диска. Ищите ISO-образы в разделе **Distros/ReactOS** на DVD – помните, их надо записывать именно как ISO, а не копировать на CD-R. Не забудьте также, что компьютер, на котором вы будете загружать дистрибутив, должен быть настроен на загрузку с CD, а не с жесткого диска; возможно, для этого придется изменить установки BIOS! ПК с процессором Pentium и 128 МБ ОЗУ прекрасно подойдет для испытания ReactOS LiveCD (его также можно эмулировать с помощью *Qemu*, см. LXF84).

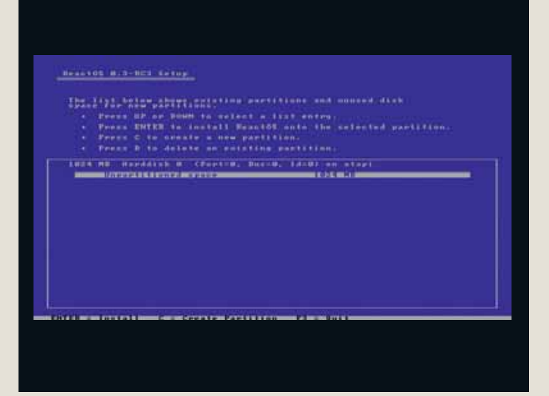
Этой весной у команды ReactOS случились трудности: один из разработчиков ОС обвинил коллег в копировании дизассемблированного кода Microsoft. Доказать ничего не удалось, но ReactOS застрял на стадии тщательной проверки кода, прекратив деятельность по разработке, из-за чего проект лишился нескольких основных участников. К счастью, аудит закончился быстрее, чем ожидалось, и ReactOS снова поздоровел. Следствием проверки стали новые правила разработки, и можно надеяться, что в обозримом будущем проекту не грозят судебные иски.

Шаг за шагом: Установка ReactOS на жесткий диск



1 Загрузка с диска

Извлеките файл **reactos0.3.0-REL-iso.zip** из раздела **Distros/ReactOS** на DVD, запишите его на CD-R и загрузите с него ваш компьютер – после подсказки нажмите любую клавишу, и у вас появится этот экран.



2 Выбор функций

Нажмите Enter, чтобы начать установку, и подтвердите настройки вашего ПК (раскладка клавиатуры, разрешение экрана и пр.). На стадии разбивки диска выберите имеющийся раздел FAT или создайте такой раздел с помощью клавиши C.

Дистрибутив Linux

PCLinuxOS MiniMe

Мало нам было дистрибутивов Linux, теперь еще и Доктор Зло появился! Ладно, шутим: MiniMe – компактная версия набирающего популярность дистрибутива PCLinuxOS (мы включали версию 0.92 на DVD LXF81). Одной из целей PCLinuxOS было преобразование Mandrake Linux (он начался как ответвление Mandrake 9.2) в дистрибутив на одном CD, содержащем лучшие программы, и создание версии Live CD, которую пользователи могут испробовать без установки на жесткий диск.

Сейчас, в версии 0.93, PCLinuxOS существует в нескольких видах: релиз Big Daddy (Большой Папочка) предусматривает широкий выбор пакетов; вариант Junior (Юниор), предоставляет базовый рабочий стол; а теперь есть и релиз MiniMe, обеспечивающий индивидуальную настройку дистрибутива с помощью малогабаритного набора пакетов по умолчанию. Построенный на ядре 2.6.16.17 и KDE 3.5.3, MiniMe загружается как Live-дистрибутив, предлагая вам опрятный и приятный рабочий стол KDE с набором приложений. И хотя функций тут не в избытке, симпатичный дизайн и четкое построение меню делают MiniMe идеальным кандидатом для демонстрации Linux потенциальным неофитам.



› Пологие холмы на заставке... где же это мы их раньше видели?



› Знаменитый Центр Управления Mandriva все еще здесь, хотя и видоизмененный.

Более того, PCLinuxOS MiniMe можно установить на жесткий диск всего несколькими щелчками мыши. Загрузившись и войдя в систему, двойным щелчком активируйте значок Install PCLinuxOS на рабочем столе, чтобы начать процесс установки. Графическая программа-инсталлятор позволит вам выделить для дистрибутива место на жестком диске – рекомендуем отвести не менее 3 ГБ, чтобы хватило для добавления других программ. После установки можете добавлять новые приложения, запустив менеджер пакетов *Synaptic package manager*, универсальный инструмент, который, возможно, встречался вам в Ubuntu.

Для запуска PCLinuxOS зайдите в раздел **Distros/PCLinuxOS-MiniMe** на диске и запишите 300-МБ файл **pclinuxosp93a-minime.iso** на CD-R (запишите его как ISO-образ, а не копируйте на диск). Загрузите компьютер (как минимум Pentium 3 с 128 МБ ОЗУ) с полученного диска, и в итоге увидите экран входа в систему. Пароль Гостя (Guest) – guest, пароль администратора (root) – root; входите в систему, как Гость, а если вы решились установить её на жесткий диск, введите пароль root. Подробности – на сайте проекта www.pclinuxos.com.

Документация

Rute

Наш опрос по поводу содержимого диска выявил ваше желание получать побольше справочной документации по программам, поэтому на сей раз мы включили сюда превосходное Руководство Rute – о работе, поддержке и администрировании систем Linux через командную строку [на английском языке, – прим. ред.]. Вы, возможно, удивитесь, с чего бы это до сих пор суетиться с командной строкой, при наличии всевозможных графических инструментов настройки, однако без знания командной строки не обойтись, если где-то произошел сбой. Научиться работать в командной строке стоит хотя бы затем, чтобы понять, как работает система Linux.

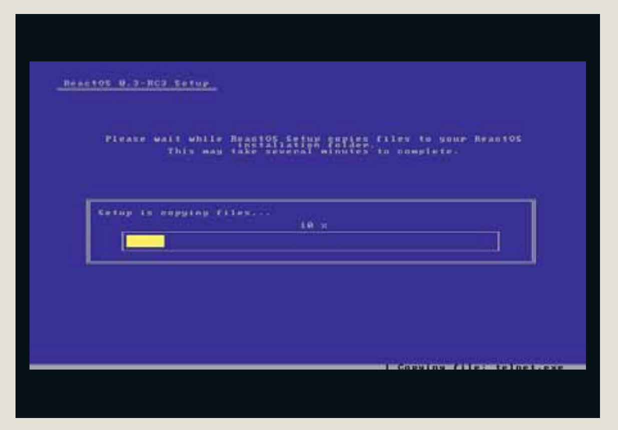
Цитируем автора: «С другими книгами вы лишь промочите ноги, а с этой вы сначала пошлепаете по мелководью, а потом нырнете с головой, но не без подачи кислорода». Так что если вы новичок или пользователь Linux среднего уровня, который почитал про оболочку в нашем журнале, но пока не очень уверен в своей способности влезть в командную строку, это – идеальное руководство для вас. Даже если вы опытный администратор, желающий просто расширить знание Linux, оно все равно вам пригодится.

4. Basic Commands
4.1 The ls Command, Hidden Files, Command-Line Options
4.2 Error Messages

› Rute выделяет команды разными цветами, это упрощает чтение.

Вы можете прочитать Rute в разделе **Help** нашего DVD. Руководство проходится по компьютерным терминам (например, «шестнадцатиричный» и «ASCII»), которые сперва кажутся не вполне по теме, но позднее вы поймете, что эти знания весьма важны для освоения Linux. Затем вас познакомят с общими командами оболочки, а за ними следует обширная сводка регулярных выражений (regular expressions) (они не такие заумные, как может показаться) и управление процессами в системе.

Rute – идеально сжатое и точное руководство по внутреннему строению Linux; если вам встретится другая документация, способная помочь читателям Linux Format – обязательно сообщите мне об этом!

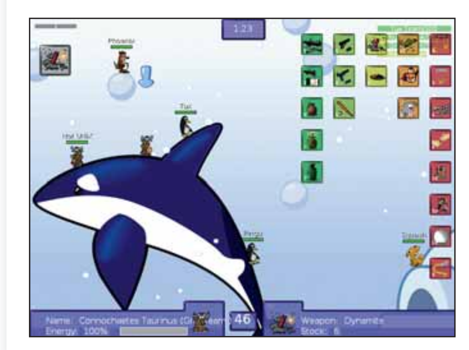


3 Копирование файлов

Снова жмите клавишу Enter – ReactOS скопируется. Выберите програму-загрузчик (MBR рекомендуется только для выделенного ПК с ReactOS!), затем можете перезагрузиться с жесткого диска и опробовать работу с приложениями

И наконец...

В заключение – несколько слов о других программах на нашем DVD: если вы пишете для Gnome или GTK, испытайте новый релиз 3.0 инструмента проектирования *Glade*. Художники и дизайнеры графики могут заняться свежими релизами *Blender* и *Inkscape*, а обладатели обширных коллекций MP3/OGG, наверное, заинтересуются *Amarok* и *Gmusicbrowser*. Тем, кто уже давно работает в Linux и не хочет отставать от последних достижений, мы предлагаем свежие версии ключевых компонентов Linux (ядро, *glibc*, *coreutils*, *Bash* и пр.) и, конечно же, игры для приятного досуга. Обратите особое внимание на *Wormux*, ради его фантастического многопользовательского режима, вдохновленного *Worms* от Team 17, и на прелестные кораблики *Bloboats* – с ними вы ощутите свежесть океанских волн на вашем рабочем столе. LXF



› Gnus захватывает Tuxen в морской битве не на жизнь, а на смерть, вчера.



Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Родион Водейко, Елена Толстякова, Александр Толстой, Иван Мищенко
Переводчики
Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабунио.
Допечатная подготовка
Мария Пучкова, Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Старший художественный редактор Мартин Парфитт (Martin Parfitt) mparfitt @futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Литературный редактор
Ребекка Смолли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрю Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

Авторы
Джонатан Остин (Jonathan Austin), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Д-р Крис Браун (Dr. Chris Brown), Энди Ченнел (Andy Channelle), Кингс Кобблер (Kings Cobbler), Бен Харлинг (Ben Harling), Энди Хадсон (Andy Hudson), Евгений Балдин, Андрей Боровский, Михаил Журухин, Дмитрий Кирсанов, Петр Семилетов, Александр Супрунов, Сергей Супрунов, Тихон Тарнавский, Александр Толстой, Алексей Федорчук, Антон Черноусов, Илья Шпаньков
Художественные ассистенты: Зигги Бейкер (Ziggi Baker), Анна Фишер (Anna Fisher), Эмит Пэтел (Amit Patel)
Фотографии: Дэвид Бленкинхорн (David Blankenhorn), Джейсон Кэплен (Jason Kaplan), Joby Sessions
Иллюстрации: Мэт Херинг (Matt Herring), Энди МакЛафлин (Andy McLaughlin), Magictorch, Отто Штейнингер (Otto Steininger), Алекс Вильямсон (Alex Williamson), Крис Винн (Chris Winn)

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Представительство в Москве:
пр.Мира, 161, телефон +7(495) 799-18-63, +7(495)136-88-45
Email: info@linuxformat.ru, Web: www.linuxformat.ru

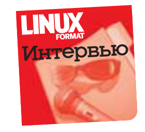


BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO • SAN FRANCISCO





В этом году Linux отмечает свое пятнадцатилетие. Присоединяйтесь – мы намерены совершить увлекательное путешествие по местам боевой славы системы, которая изменила мир. Плюс: мнения светил Open Source о том, куда движется Linux и что его ждет.



Нат Торкингтон

Редактор O'Reilly, организатор OSCon и глава проекта Perl 6.



На диске:

Slackware, Gentoo и эксклюзивные материалы из архива Linux Format!

Внимание! Содержание следующих выпусков может меняться без уведомления.

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» 6 номеров (июль, август, сентябрь, октябрь, ноябрь, декабрь 2006 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2006 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2006 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

- по выставленному счету (для юридических лиц)
- по квитанции в любом отделении Сбербанка

Плюсы подписки

- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2007!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ НА ЖУРНАЛ

Linux Format

ИНДЕКС ИЗДАНИЯ

КОЛИЧЕСТВО КОМПЛЕКТОВ

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА

ПОЧТОВЫЙ ИНДЕКС — АДРЕС ДОСТАВКИ

КОМУ

ФАМИЛИЯ, ИНИЦИАЛЫ

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ — МЕСТО — ЛИТЕР

НА ЖУРНАЛ

ИНДЕКС ИЗДАНИЯ

Linux Format

НАИМЕНОВАНИЕ ИЗДАНИЯ

СТОИМОСТЬ: ПОДПИСКИ — РУБ. — КОП.; ПЕРЕАДРЕСАЦИИ — РУБ. — КОП.

КОЛИЧЕСТВО КОМПЛЕКТОВ

НА 2007 ГОД ПО МЕСЯЦАМ

1	2	3	4	5	6	7	8	9	10	11	12

КУДА

ПОЧТОВЫЙ ИНДЕКС — АДРЕС ДОСТАВКИ

КОМУ

ФАМИЛИЯ, ИНИЦИАЛЫ



ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса» – совместный проект Государственного предприятия «Казпочта», Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан
- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения
- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия
- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия
- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан
- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия
- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул.Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан
- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина
- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22

ПОДПИСКА НА LINUX FORMAT

Агентство “Centerpress”

Сколько стоит подписка?
Подписка на журнал "Linux Format" 12 номеров (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь 2007 года) стоит 1800 рублей.

Как оформить подписку?
Чтобы оформить подписку на журнал "Linux Format", необходимо зарегистрироваться в интернет-агентстве Centerpress.ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар "Подписка на журнал "Linux Format" на 2007 год 12 номеров (01-12 / 2007)", получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Агентство “Centerpress”: www.centerpress.ru

Все Плюсы подписки!
- Подписка дешевле!
- Гарантированное получение журнала!

По каталогам РФ

Каталог агентства “РОСПЕЧАТЬ” - подписной индекс

20882

Каталог “ПРЕССА РОССИИ” - подписной индекс

87974



АЛЬТЕРНАТИВНЫЕ АГЕНСТВА РФ

Агентство «Интер-Почта» (095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса» (095) 787-34-47

Агентство «УралПресс»

- Екатеринбург, Березовский, В. Пышма, Первоуральск тел. (343) 375-80-71, 375-84-93, 375-84-39, факс 375-62-74, info@ural-press.ru
- Нижний Тагил тел. (3435) 411448, 417709, ntagil@ural-press.ru
- Челябинск тел. (351) 262-90-03, 262-90-05, pochta@chel.surnet.ru
- Пермь тел. (3422) 60-24-40, 60-22-95, 60-35-42, parma-press@permonline.ru